Kinetic API Feature/Change request

(Power Management)

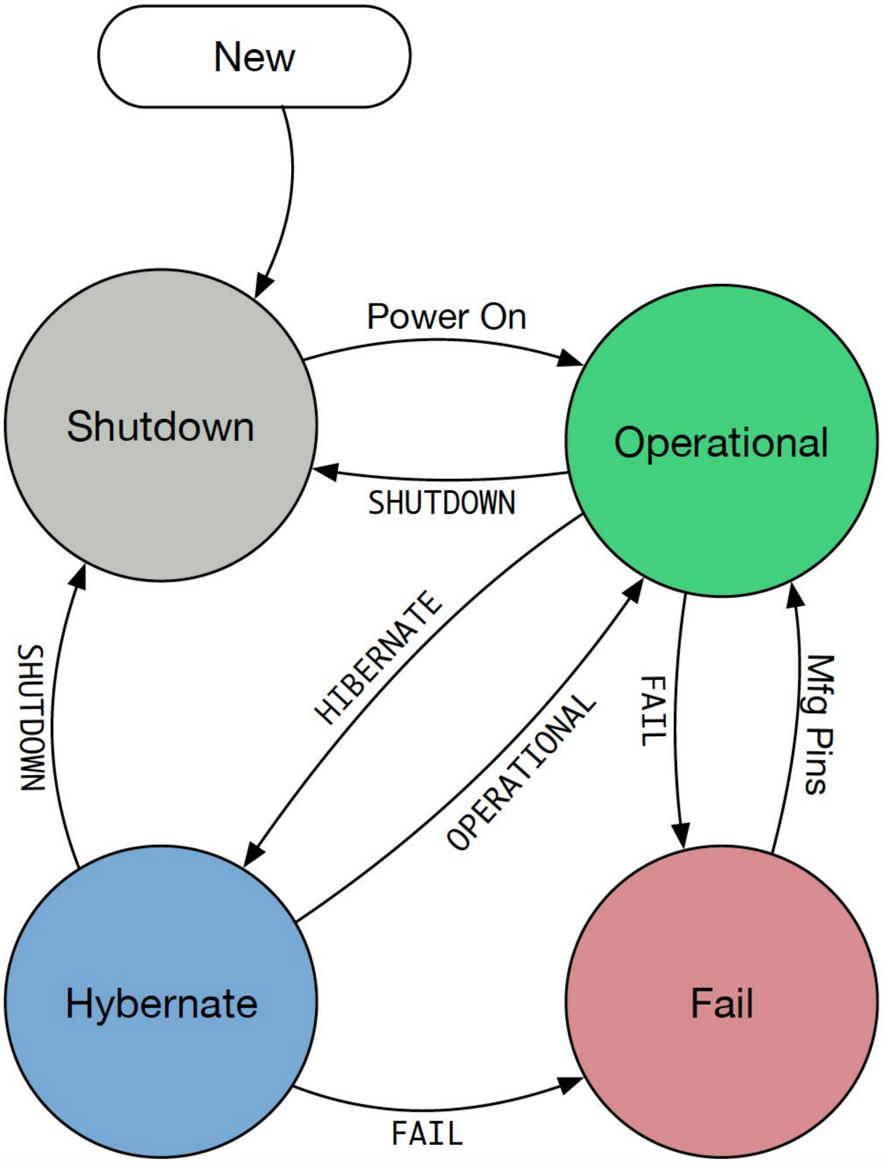
Description of required functionality (individual command syntax and explanation of expect results):

Business clients expect to be able to place the drives in lower power states (or even turn them off completely).

This feature proposal provides a mechanism for changing between power states, observing the current power state and access control for such feature.

Test	Library	Simulator	Device
X	X	X	X

Affected code bases (check all that apply)


New
Shutdown
Power On
Operational
SHUTDOWN
HIBERNATE
SHUTDOWN
OPERATIONAL
FAIL
Mfg Pins
Hybernate
Fail
FAIL

The Power Management operation

```
enum MessageType {
        SET_POWER_LEVEL = 48;
        SET_POWER_LEVEL_RESPONSE = 47;
}

message Body {
        optional PowerManagement power = 10;
}

message PowerManagement {
        optional PowerLevels level = 1;
}
```

There are currently four distinct power states we can differentiate at the system level:

1. Fully operational, running at full capacity.

2. Hibernation, which puts the drive on a very low power mode where all operations other than changing power state are rejected. Transitioning to hibernation will allow all pending operations to terminate normally, dirty caches flushed, and all new non-power management commands to be rejected with HIBERNATE_EXCEPTION. Drive shall cache keys for Identities that are allowed to do power management (This means that hibernate will be before hmac calculations.)

3. Shutdown, where the drives is turned off, only way to get it back up is a physical power cycle in the drive. Transitioning to shutdown will allow all pending operations to terminate normally dirty caches flushed, and connections to be terminated with unsolicited status SHUTDOWN_EXCEPTION.

4. Fail. If the drive is broken beyond repair, command the drive to power off permanently. Even is the drive power is cycled, do not power up to even to load the OS. There is no way for the customer to bring the drive back, Device Vendor can interrupt the boot sequence through the manufacturing pins to restore the drive to operational. The drive should attempt a graceful shutdown where the dirty caches are flushed, but if unable to accomplish the flush, the drive should still enter the fail state.

```
enum PowerLevels {
        INVALID_LEVEL = -1;
        OPERATIONAL = 1; // Fully operational mode, default mode for the drive
        HIBERNATE = 2; // All operations will be rejected
        SHUTDOWN = 3;
        FAIL = 4;
}
```

Exception management

When a drive is in hibernation, and receives an operation that is not power-up, the operation must be rejected and the power level must remain unchanged.

We need a new status codes for the power states.

```
enum StatusCode {
        HIBERNATE = 22;
        SHUTDOWN = 23;
}
```

Security

Access control to changing power levels must be managed independently of other operations.

```
enum Permission {
        POWER_MANAGEMENT = 9;
}
```

Requesting current power level

The unsolicited GetLog at the beginning of a connection will return the drive status. The only two possible values are OPERATIONAL or HIBERNATE.

```
message Configuration {
	optional PowerLevels current_power_level = 18;
}
```

Compatibility

This is an incremental feature with no backward compatibility issues.

Upgrading

Standard process.

How to validate/test feature to verify correct implementation:

Java API and simulator will implement this feature as reference implementation. Conformance Test Suite will have test cases to validate this new feature.

Value of feature/alternatives:

It is a high priority feature for large customers which can’t possible power their entire storage infrastructure simultaneously.

Additionally it provides energy savings for cold storage customers.