Chris Long

[Email address]

**CDRL: A009 Scientific and Technical Reports**

Internal Testing and Evaluation/Interim Results

Contact No: N6600117C2001

Prepared by: Next Century Corporation

2701 Technology Drive Suite 100

Annapolis Junction, MD 20701

WARNING: This document contains technical data whose export is restricted by the Arms

Export Control Act (Title 22, U.S.C., sec. 2751, et seq.) or the Export Administration Act of

1979, as amended, Title 50, U.S.C., App. 2401 et seq. Violation of these export laws are

subject to severe criminal penalties. Disseminate in accordance with the provisions of DoD

Directive 5230.25.

SAVIOR System Interface Details

Version 2, January 2019

# Introduction

This document describes the external interfaces to the SAVIOR system.

# Client Interfaces

SAVIOR is designed to interact with three client applications: the Admin Workbench, Dashboard, and Desktop.

The Admin Workbench and Dashboard are web applications. They interact with rest of the SAVIOR system using the Virtue Management API (see §3).

The Desktop is a Java application running in a typical desktop environment (e.g., Windows, Linux). It also uses the Virtue Management API. In addition, it connects to Linux application VMs (running as Xen DomUs) via ssh and xpra[[1]](#footnote-1) (tunneled over ssh) and Windows application VMs (running as EC2 instances) via ssh and RDP.

# Virtue Management API

The Virtue management API implmented in SAVIOR is based on the API published by APL. The current version of SAVIOR does not implement the entire API as specified by APL.

The SAVIOR API also has a structural difference from the APL API, in the relationship between applications and Virtues. In the APL API, Virtues directly contain applications. However, in SAVIOR, Virtues contain virtual machines that contain applications.

The Virtue Manager (the “virtue-admin” module in the source code) provides a REST service implementation of the Virtue APIs. The following sections describe the supported calls. Unless otherwise indicated, parameters are passed as part of the URL and denoted by curly braces ({}).

In test mode, user authentication is not enforced. When authentication is enforced, all API calls require a user authentication token set as the cookie “JSESSIONID”, except for User Login, which returns the token when successful (see POST /login in SystemInterfaceSpecification.pdf).

Currently supported operating systems are Linux and Windows, output as “LINUX” and “WINDOWS”.

The Date type is a string in ISO 8601 date and time format (e.g., “2018-04-16T01:22:07.455+0000”).

## Virtue User API

See SystemInterfaceSpecification.pdf for the Virtue and User API.

## Virtue Admin API

See SystemInterfaceSpecification.pdf for the Virtue and User API.

# Sense and Response API

The Sense and Response API is described in its own document.

1. https://xpra.org/ [↑](#footnote-ref-1)