

|  |
| --- |
| APS  TMT.CTR.TEC.??.???.DRF01 |

February 21, 2017

# 

TABLE OF CONTENTS

1 Introduction 3

1.1 Audience 3

1.2 Scope 3

1.3 Applicable Documents 3

1.4 Reference Documents 3

1.5 Change Record 3

1.6 Acronyms 3

2 Architecture 3

3 Assembly Design 3

4 HCD Design 3

5 engineering UI Design 3

6 Software Development Environment 4

7 Deployment Architecture 4

# Introduction

TBD.

## Audience

This document is targeted primarily towards software developers (TBD).

## Purpose

This document is the principle design documentation for the APS Stimulus Prototype software. This document will also be a key source of input to the preliminary design of the APS ICS system, particularly in the areas of CSW integration and component design with emphasis on motion control. The corresponding software development effort and technology choices will also inform similar choices for the APS ICS preliminary design.

## Scope

This document provides the design basis for the APS Stimulus Prototype based on TMT Common Services and the APS Stimulus Software requirements. This document will evolve as more information becomes available, particularly in the area of HCD/Controller design, which will become available sometime after the first draft.

## Applicable Documents

1. [Design Requirements Document for OSW Common Software](https://docushare.tmt.org/docushare/dsweb/Get/Document-12130), TMT.SFT.DRD.08.002.
2. <add the optomechanical design docs>

## Reference Documents

1. No Ref Docs.
2. <add CSW SDD>

## Change Record

|  |  |  |  |
| --- | --- | --- | --- |
| Revision | Date | Who | Modifications |
| *DRF01* | 2/10/17 | SM | Initial Draft |
|  |  |  |  |
|  |  |  |  |

## Acronyms

**API** Application Programmer Interface

**CS** Configuration Service

**CSW** Common Software

**DRD** Design Requirements Document

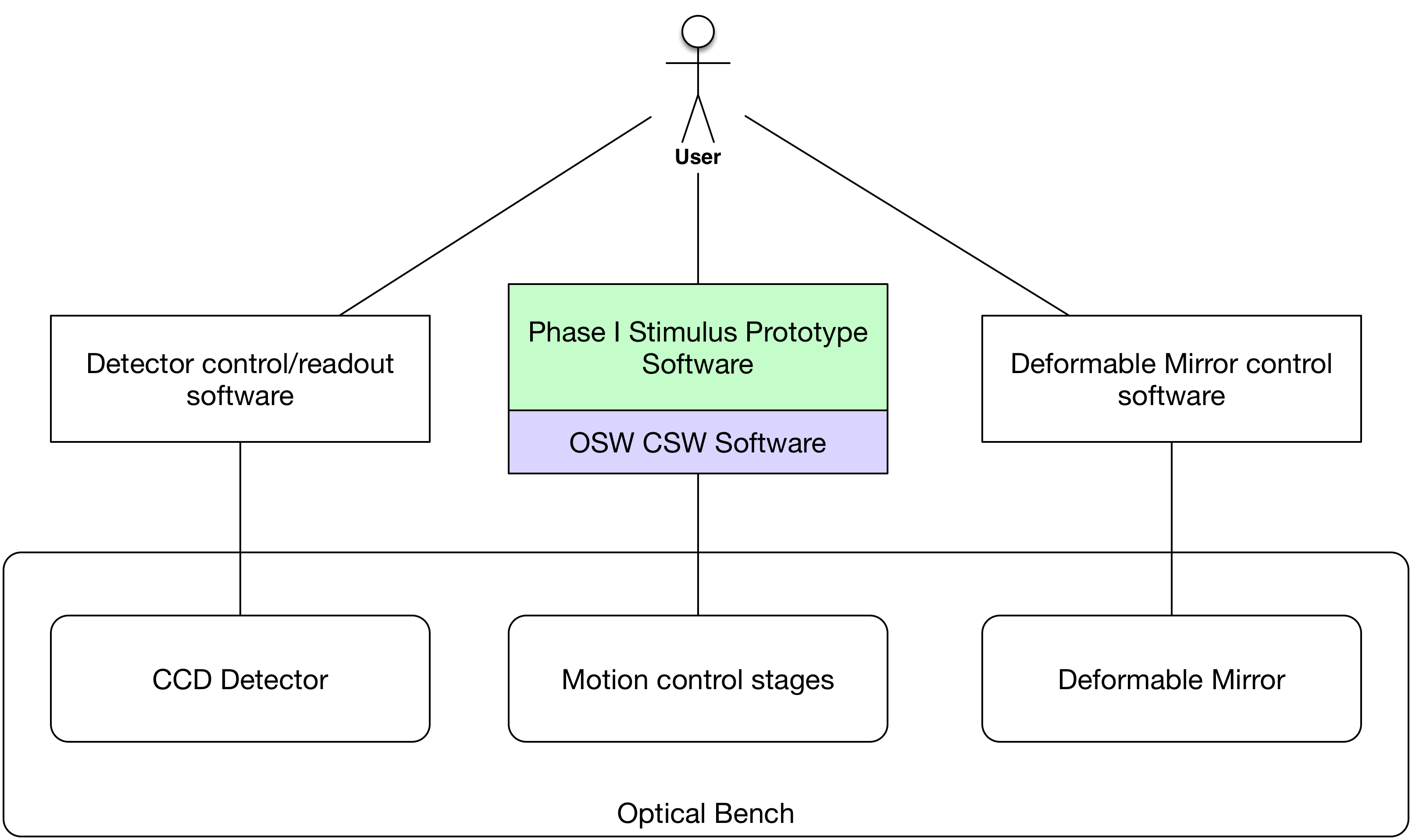
**OSW** Observatory Software

**TMT** Thirty Meter Telescope

# Overview

## System Context

The Stimulus Prototype software will control four motion control stages within the stimulus. Stand-alone software will be used to control the stimulus deformable mirror itself and to collect images from a detector.

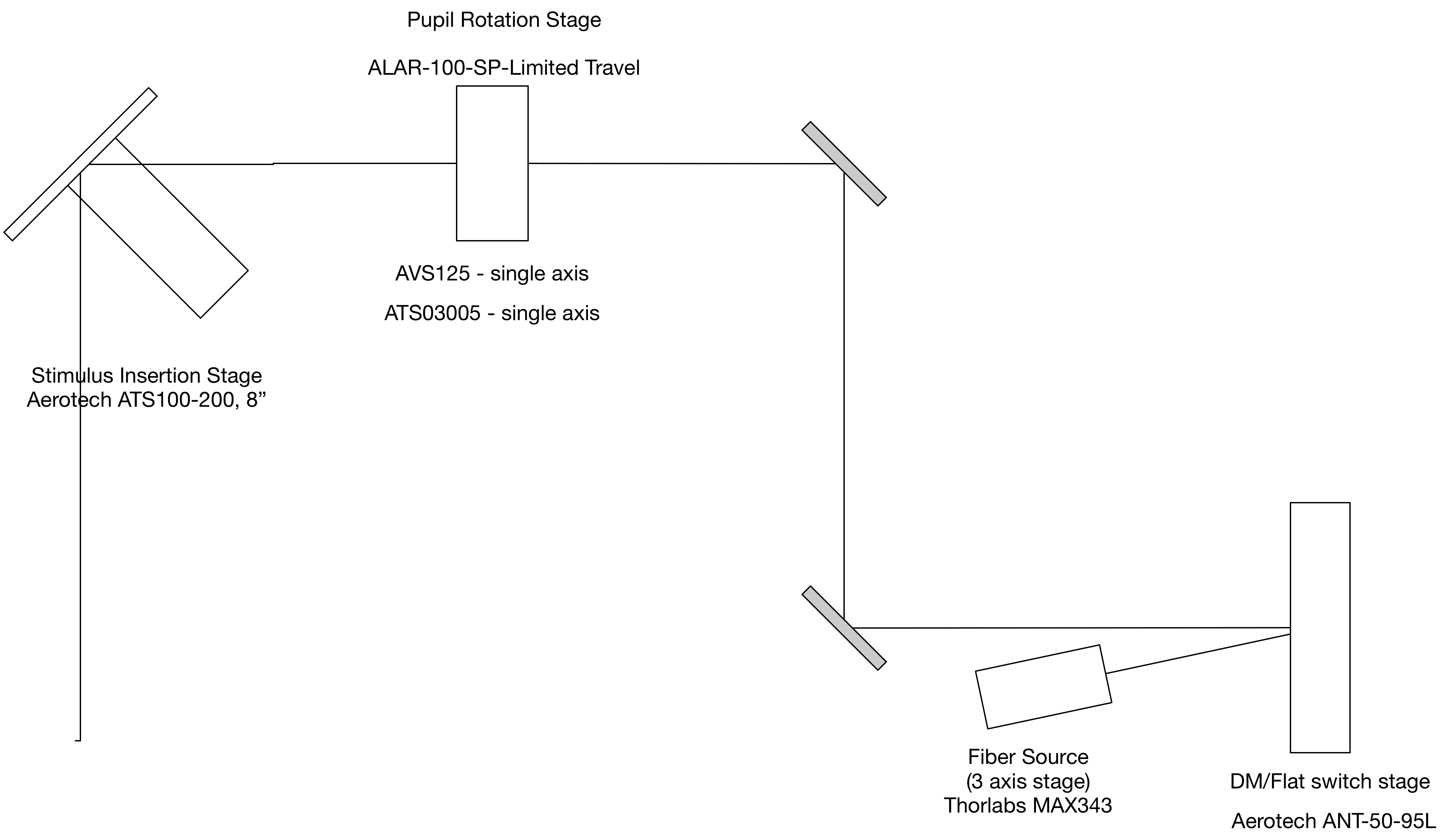


A user of the stimulus prototype will use three software systems in Phase I: the prototype software which will use CSW software libraries and follow CSW design practices, the detector control/readout software (TBD), and the deformable mirror control software provided by the deformable mirror vendor.

In a future phase it will be desirable to include detector control software into the prototype software, as this activity will be key in developing a preliminary design for ICS components related to detectors.

### Bench Layout of Stages

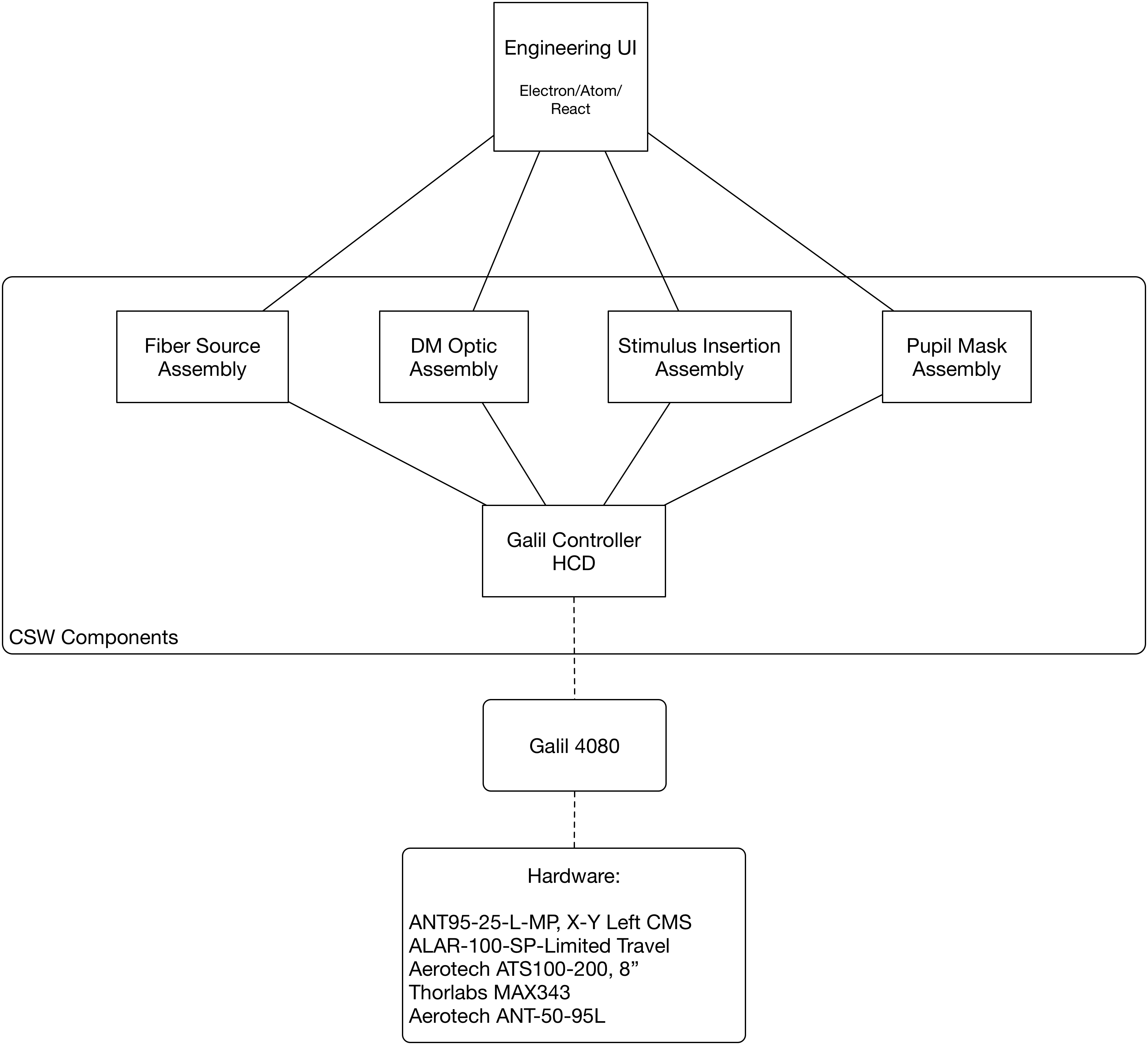
The following diagram shows the four stages in the context of the optical bench (not scaled) and showing the optical path.



The stimulus light source is a fiber source on a 3-axis Thorlabs stage. The beam is reflected off either the deformable mirror or a flat mirror, which can be switched between using the Aerotech ANT-50 stage. The beam continues to the pupil rotation stage, which is also a three-axis stage. The final stage is a two-position insertion stage, which will be used to switch between light from the telescope and light from the stimulus to the rest of the APS.

More detailed information on the bench optomechanical design can be found in TBD.

## Logical Software Architecture



## Physical Software Architecture

## Assumptions/Constraints

The design will make the most use possible of CSW, so that its alignment with APS-ICS needs can be evaluated and uncover any issues with the CSW design itself.

The implementation will be in the Scala programming language, so that the team may become more familiar with it, and to determine if it is a better choice than Java for APS-ICS.

User Interface technologies used for the engineering user interface will be chosen based on current OSW preferences and current market trends, potentially filling gaps in TMT knowledge of available products.

## Items Not Included

Hardware controller software design is not included in the first draft. As the team acquires experience in this area, a design will be developed.

HCD design is not included in the first draft. The HCD design is dependent on the hardware controller software design.

Detector software design is not included in this development phase of the prototype.

# Conceptual Design

## Components/Lifecycle

## Commands

## Telemetry

## Alarms

## Logging

## Configuration

## Engineering User Interface

# Assembly Design

TBD

## Assembly Actors/Classes

## Assembly Actors Message Passing

# HCD Design

TBD

# Controller Software Design

TBD

# engineering UI Design

## Screen Mock-ups

# Testing

# Software Development Environment

TBD.