

# **System Design Document**

---

**Chair for Applied Software Engineering, TUM**



# Table of Contents

---

<b>1 System Design Document</b>	
1.1 Introduction	1
1.1.1 Purpose of the system	1
1.1.2 Design goals	1
1.1.3 Definitions, acronyms, and abbreviations	2
1.1.4 References	2
1.2 Current software architecture	3
1.3 Proposed software architecture	4
1.3.1 Overview	4
1.3.2 Persistent data management	4
1.3.2.1 Audio	4
1.3.2.2 Video	4
1.3.2.3 Tracking	5
1.3.2.4 Orchestra	5
1.3.2.5 UI	5
1.3.2.6 Distribution of data	5
1.3.3 Hardware/software mapping	5
1.3.3.1 Components	5
1.3.3.2 Component Diagrams	6
1.3.3.3 Nodes	6
1.3.3.4 Deployment Diagrams	6
1.3.4 Access control and security	7
1.3.5 Global software control	7
1.3.5.1 Sequence Diagrams	7
1.4 Subsystem services	15
1.5 Glossary	16



# 1 Introduction

---

The purpose of this section is to provide a brief overview of the software architecture and the design goals. It also provides references to other documents and traceability information (e.g. related requirements document, references to existing systems, constraints impacting the software architecture).

## Sections:

- Purpose of the system
- Design goals
- Definitions, acronyms, and abbreviations
- References

## 1.1. Purpose of the system

## 1.2. Design goals

### Design Goal: The system must be extensible

it must be easy to add new functionality or new classes of a subsystem without having a large impact on the rest of the subsystems, which can be achieved by minimizing coupling and maximizing coherence

### Design Goal: The system must be scalable up to eighty musicians

### Design Goal: The System should support Tempi from Largo to Presto

Comments	Description
Actual Tracking Performance	Tracking tests conditions: 1:Background: While scanning only the background no points shall be found 2:2m distance from baton to camera 3:Area available for conducting: width 1.6m height 1.2m Test results: Maximum conducting speed: 120 bpm

### Design Goal: Users can use the system with a low- to mid-cost camera

Open Action Items	Description
Add all identified design goals to the Design goals section	Please add all design goals of the presentation to Sysiphus.

### **1.3. Definitions, acronyms, and abbreviations**

### **1.4. References**

## 2 **Current software architecture**

---

The purpose of this section is to make explicit the background information that system architects used, their assumptions, and common issues the new system will address.

## 3 Proposed software architecture

---

This section documents the system design model of the new system.

### Sections:

- Overview
- Persistent data management
- Hardware/software mapping
- Access control and security
- Global software control

### 3.1. Overview

This chapter presents a bird's-eye view of the software architecture and briefly describes the assignment of functionality to each subsystem.

### 3.2. Persistent data management

Persistent data management describes the persistent data stored by the system and the data management infrastructure required for it. This section includes the description of data schemes, the selection of a database, and the description of the encapsulation of the database.

### Sections:

- Audio
- Video
- Tracking
- Orchestra
- UI
- Distribution of data

#### 3.2.1. Audio

The audio component stores .aif (audio interchangeable file format) files.

bit rate: 705 kbps

sample rate: 44.100 kHz

channels: mono

encoding type: linear PCM (pulse code modulation)

the system supports any common fileformat (mp3, aif, aac, wav, etc)

#### 3.2.2. Video



The video component stores quicktime movies.

### 3.2.3. Tracking

The tracking component stores a gesture reference database and a conducting session history, both as mysql databases.

### 3.2.4. Orchestra

The orchestra component stores encyclopedia information using a mysql database. It also stores basic music configuration via an XML file and keeps personalized music packages as persistent objects using an XML file.

### 3.2.5. UI

The UI component stores the last used configuration as a persistent object inside a configuration file.

### 3.2.6. Distribution of data

Data should be distributed within the hardware according to the local distribution of the components using the data.

## 3.3. Hardware/software mapping

Here will be described how subsystems are assigned to hardware and off-the-shelf components. It also lists the issues introduced by multiple nodes and software reuse.

#### Sections:

- Components
- Component Diagrams
- Nodes
- Deployment Diagrams

### 3.3.1. Components

#### Component: AudioController

<i>Depending Components</i>	Orchestra
<i>Related Object Models</i>	AudioController

#### Component: Orchestra

<i>Needed Components</i>	AudioController, VideoController, TrackingController
<i>Depending Components</i>	UI

### Component: TrackingController

<i>Depending Components</i>	Orchestra
<i>Related Object Models</i>	TrackingController

### Component: UI

<i>Needed Components</i>	Orchestra
<i>Related Object Models</i>	UserInterface, MainController, AdminController, AdminPanel

### Component: VideoController

<i>Depending Components</i>	Orchestra
-----------------------------	-----------

## 3.3.2. Component Diagrams

### Component Diagram: Overview

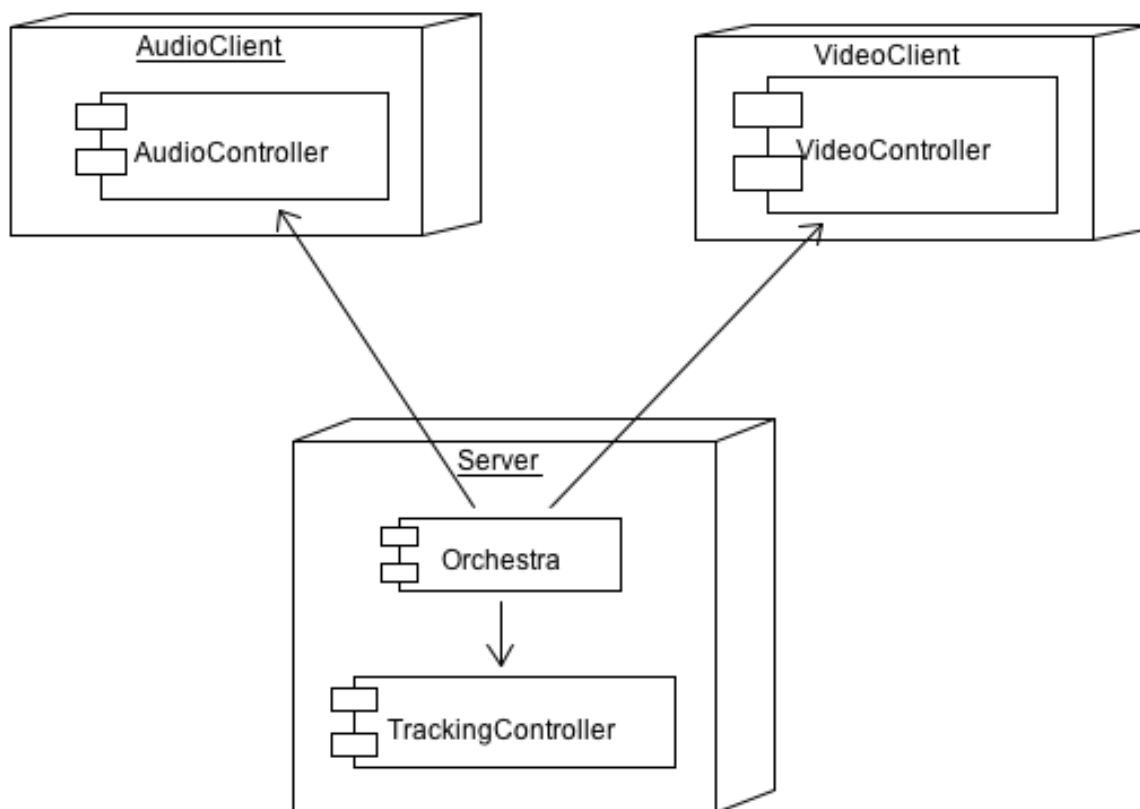
### 3.3.3. Nodes

#### Deployment Node: AudioClient

#### Deployment Node: Server

#### Deployment Node: VideoClient

## 3.3.4. Deployment Diagrams



Deployment Diagram: DistributedAudio

### 3.4. Access control and security

Here the user model of the system in terms of an access matrix will be described. This section also documents security issues, such as the selection of an authentication mechanism, the use of encryption, and the management of keys.

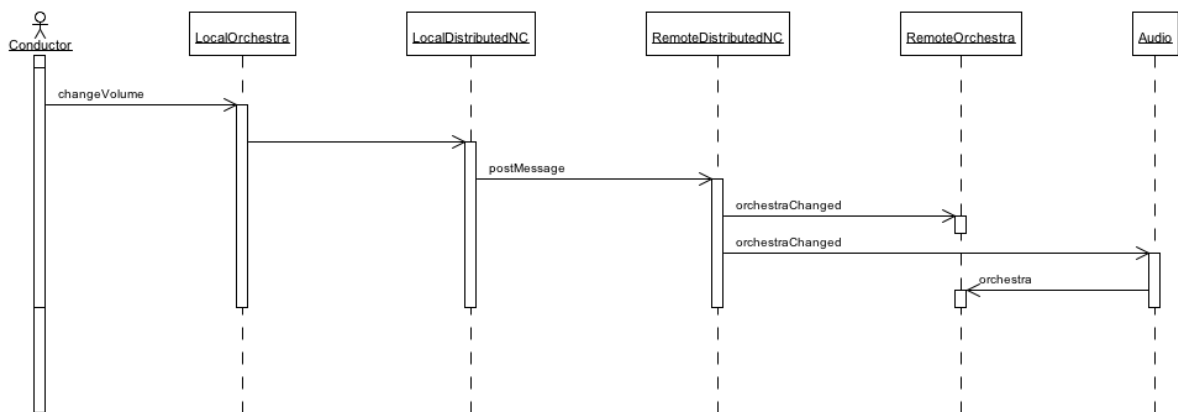
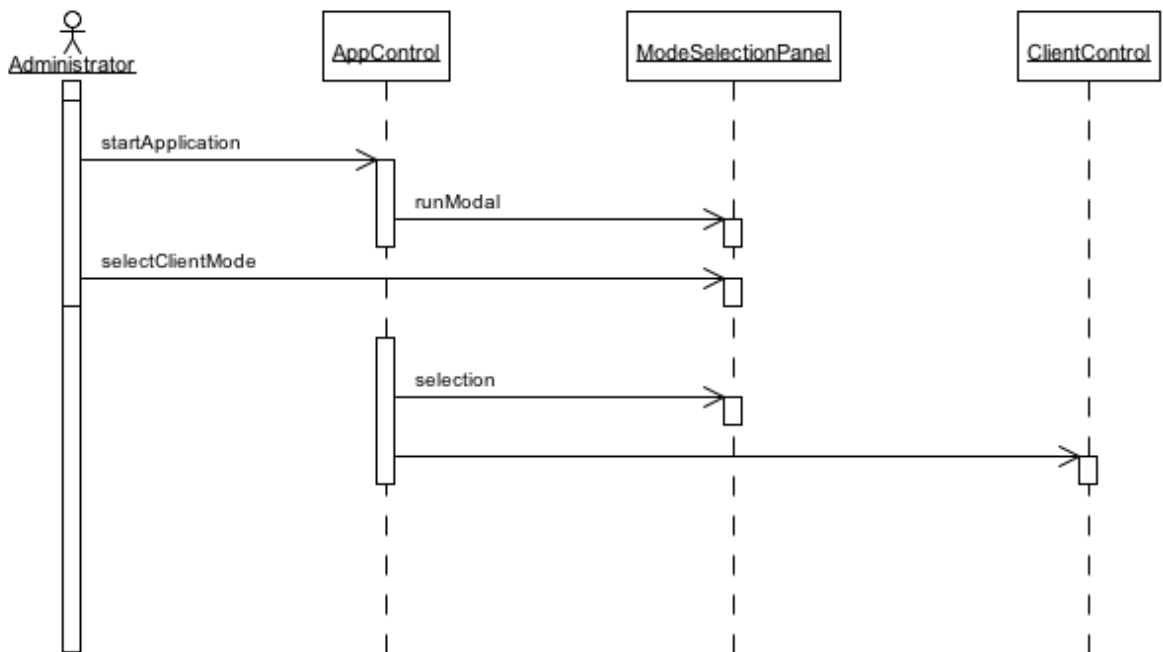
### 3.5. Global software control

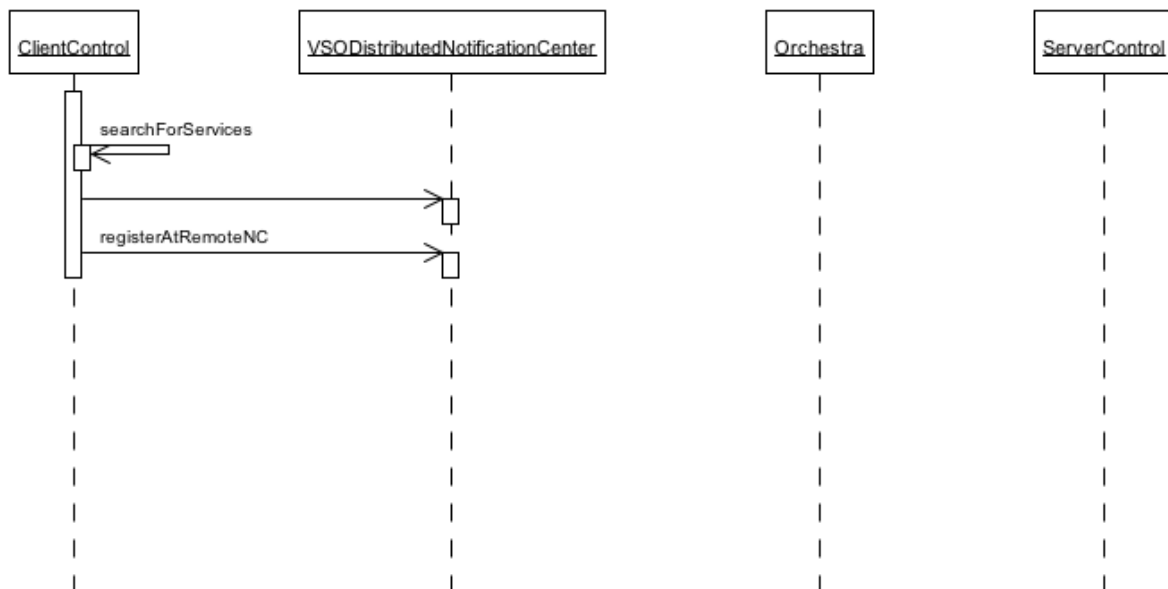
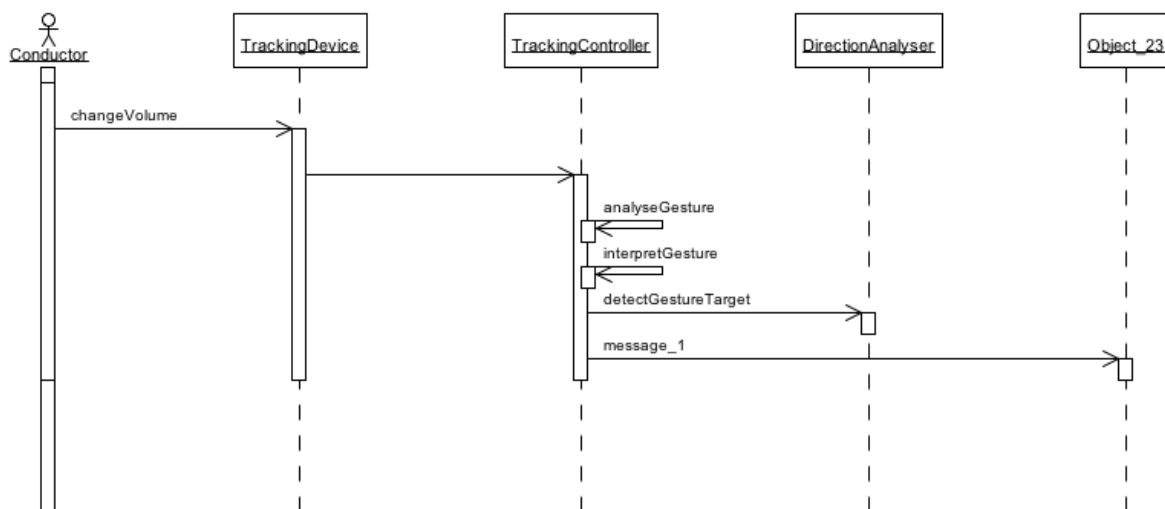
It describes how the global software control is implemented. In particular, this section describes how requests are initiated and how subsystems synchronize. That means that synchronization and concurrency issues will be listed and addressed in this chapter.

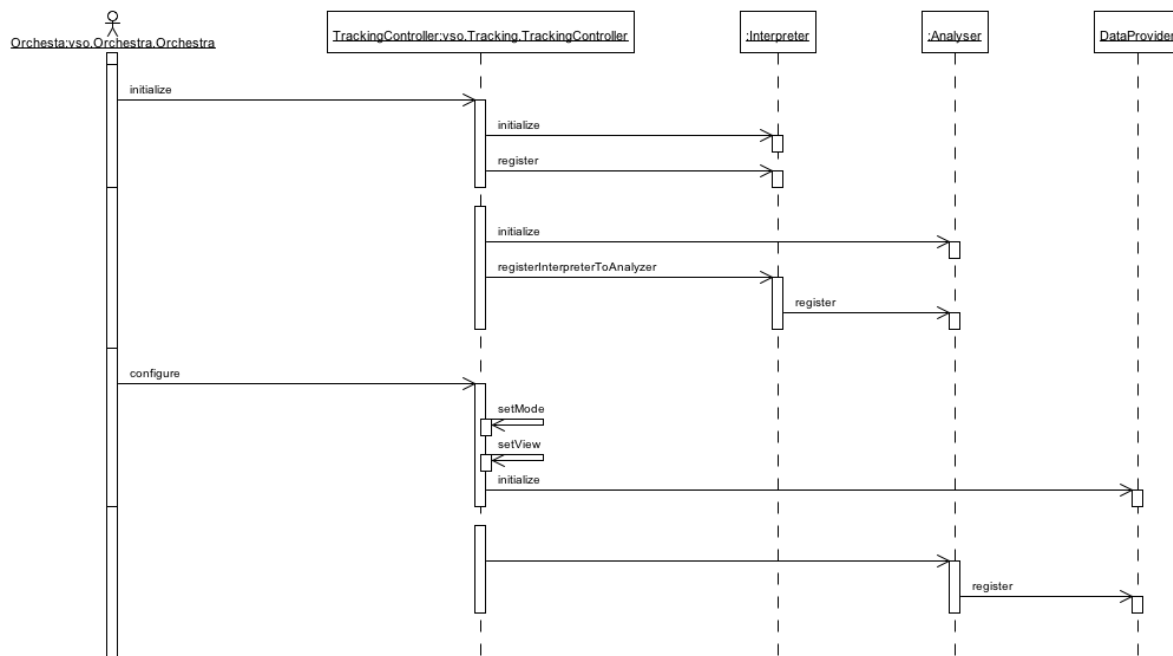
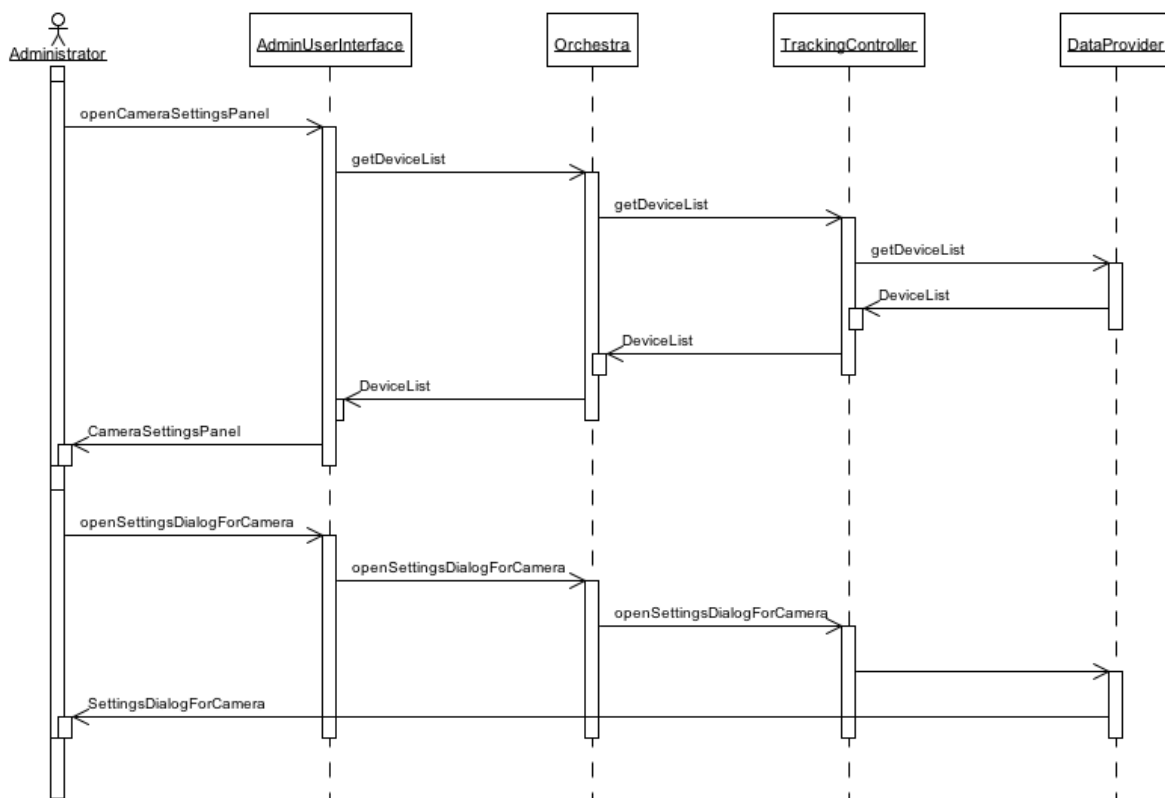
#### Sections:

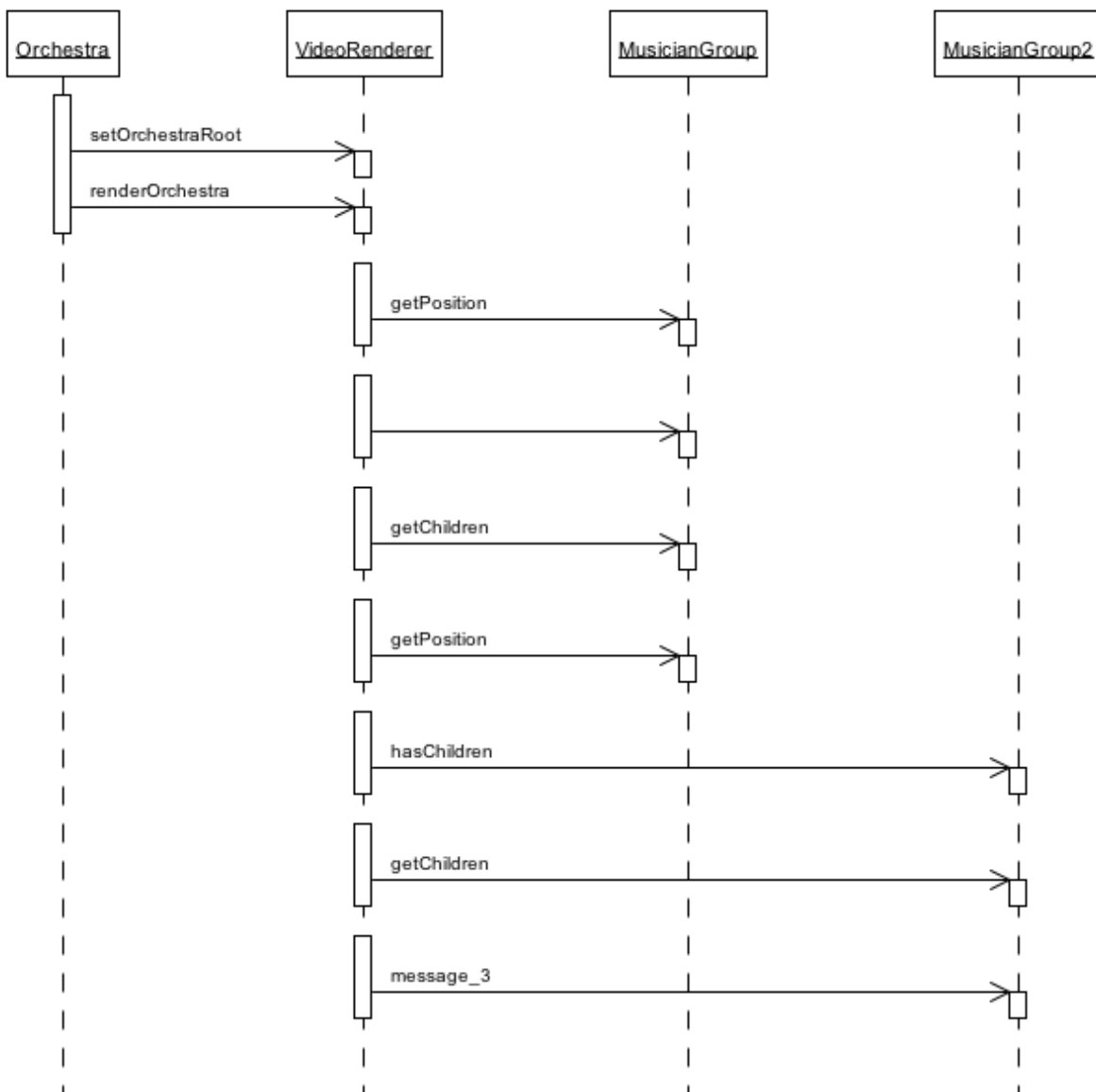
- Sequence Diagrams

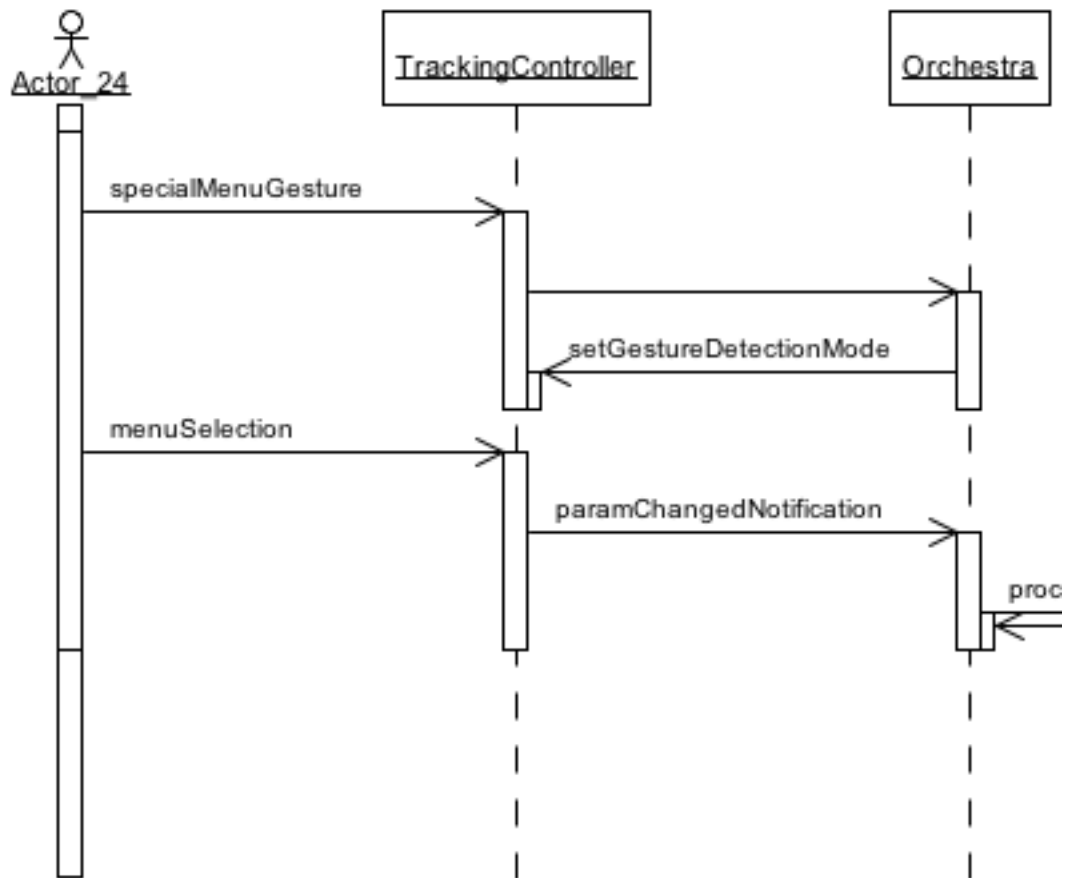
#### 3.5.1. Sequence Diagrams



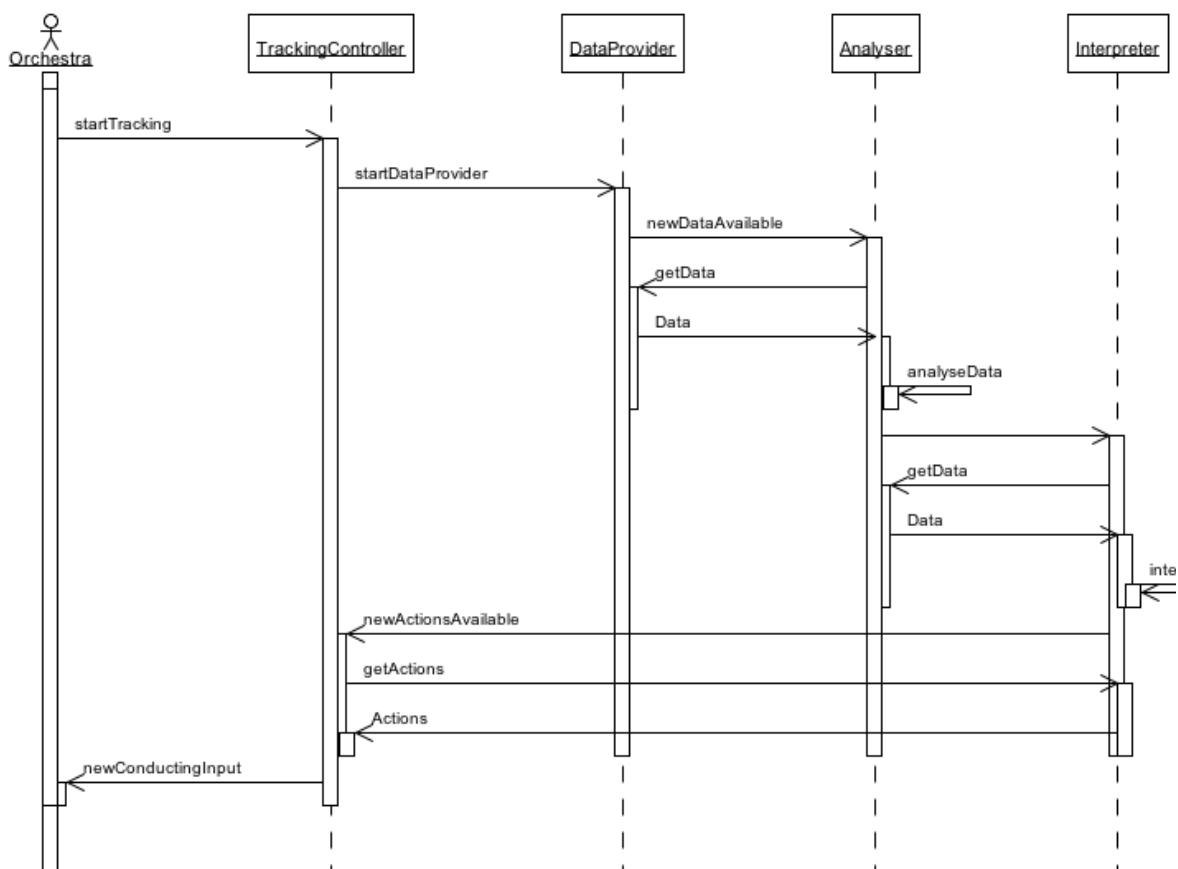
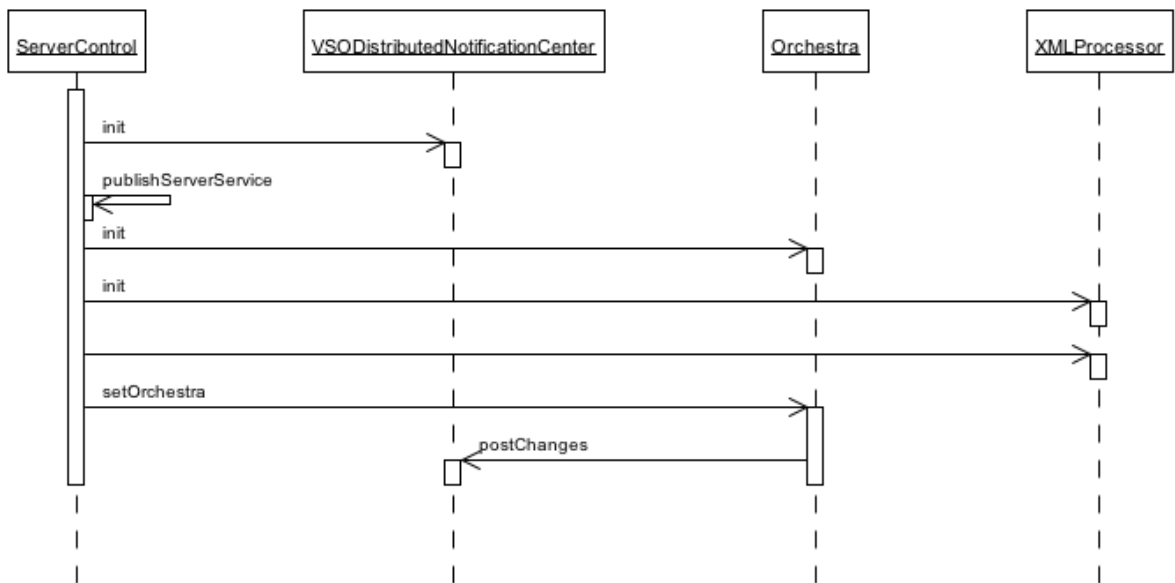












**Sequence Diagram: ApplicationStart****Sequence Diagram: ChangesNotification****Sequence Diagram: ChangingVolume****Sequence Diagram: ClientStart****Sequence Diagram: ConfigureTrackingCameras**

Describes how the orchestra gets the settings dialog for each camera, which is provided by QuickTime. This settings dialog should get delegated to UI.

**Sequence Diagram: InitializeTracking**

Describes the initialization and configuration of the tracking subsystem by the Orchestra subsystem for a new conducting session

**Sequence Diagram: RenderOrchestra**

The orchestra instance calls the function renderOrchestra from the video api, the video controller walks through the orchestra objects and renders the current frames.

**Sequence Diagram: SelectMenuItem****Sequence Diagram: ServerStart****Sequence Diagram: TrackingSession**

## 4 Subsystem services

---

This section describes the services provided by each subsystem in terms of operations. This chapter should serve as a reference for teams for the boundaries between their subsystems. The interface of each subsystem can be derived from this section and detailed in the Object Design Document.

## 5 **Glossary**

---