**Version Description Document (VDD)**

**Recaster**

**ver. 2.3.1**

**Prepared by:**

Alex Chernilov, Impleotv Systems Ltd.

**Table of Contents**

[Scope 2](#__RefHeading___Toc931_1150163978)

[Identification 2](#__RefHeading___Toc933_1150163978)

[System Overview 2](#__RefHeading___Toc935_1150163978)

[Documentation Overview 3](#__RefHeading___Toc937_1150163978)

[Referenced Documents 3](#__RefHeading___Toc939_1150163978)

[Version Description 3](#__RefHeading___Toc941_1150163978)

[Inventory of Materials Released 3](#__RefHeading___Toc943_1150163978)

[Inventory of CSCI Contents 4](#__RefHeading___Toc945_1150163978)

[Summary of Changes 4](#__RefHeading___Toc947_1150163978)

[Installation Instructions 4](#__RefHeading___Toc949_1150163978)

# Scope

## Identification

This document is the Version Description Document (VDD) for the Recaster software

This document describes the Version 2.3.1

## System Overview

The Stream Recasting Web Application is a client-server architecture designed to receive UDP (unicast or multicast) streams and transmit them to one or more targets while offering optional protocol conversion. This documentation will guide you through the setup and usage of this application, outlining its key features, and explaining how to control and monitor the streams.

Key Features:

- Stream recasting (UDP in -> UDP / TCP out).

- Optional protocol conversion for outgoing streams.

- Monitoring of input stream status and bitrate.

- YAML-based configuration for platform and channel settings.

- REST API for managing platforms, channels, and their behavior.

- React frontend for user control and monitoring.

**Client-Server Architecture**

The system follows a client-server architecture. The backend handles stream recasting, input stream monitoring, and configuration management. The frontend serves as the user interface for controlling and monitoring the streams.

**Backend**

The backend reads the initial configuration from a YAML file, 'platforms.yml', and provides a REST API for adding, editing, and deleting platforms, channels, and outputs. It manages the stream recasting process and input stream monitoring.

The backend spawns a separate process for every output channel to leverage multiple CPUs and improve robustness. This approach enhances the system's performance and resilience.

**Frontend**

The React frontend provides a web-based interface for users to interact with the system. It allows users to control platforms and channels and provides monitoring capabilities for input streams.

**Monitoring Features**

The frontend provides monitoring features for input streams, displaying information such as status and bitrate, enabling users to make informed decisions regarding stream management.

## Documentation Overview

This documentation provides an overview of the Stream Recasting Web Application and its various components for specific version. For detailed instructions on installation, configuration, and usage, refer to the project-specific documentation and user guides.

# Referenced Documents

Recaster manual. Provided as an interactive SW.

# Version Description

## Inventory of Materials Released

The materials released include this document provided along the SW version 2.2.1

## Inventory of CSCI Contents

The application is provided as an archive file and includes the runtime files and the corresponding manual.

## Summary of Changes

Version related changes can be seen in the CHANGELOG.md file in the root directory.

## Installation Instructions

**Prerequisites**

Before setting up the application, ensure that you have [Node.js](https://nodejs.org/en) installed:

**Installation**

The recaster application is cross-platform and does not require any additional dependencies apart from Node.js.

The Recaster binaries can be downloaded from the github repository. The latest version is available from the direct download link shown on the main (README) page. Previous releases can be found in the Releases section. The binaries are available in both tar.gz and zip formats.

**Configuration**

The configuration settings for the application are stored in the /db folder. Following the installation (or if the internal database does not exist or is empty), Recaster will search for a **platforms.yml** file in the **/data** folder.

It defines the high-level platform entity and the associated channels, inputs, and outputs.

The **platforms.yml** file will be loaded to create the initial configuration. Before loading the file, Recaster performs a validation check to ensure its validity. The application will only proceed to load the file if it meets the required criteria.

Note, it is important to follow the [yaml format rules](./user-guide/yaml-guidelines.md)

More on the [platforms.yml file](./platforms-config-yml.md) .

After the initial configuration is set up, the recaster will store its configuration in the database, creating a dynamic and responsive configuration environment for the application. Any subsequent changes to the configuration will be reflected in updates to the platforms.yml file. This guarantees that the configuration file remains synchronized with the latest settings, making it easy to back up or share.