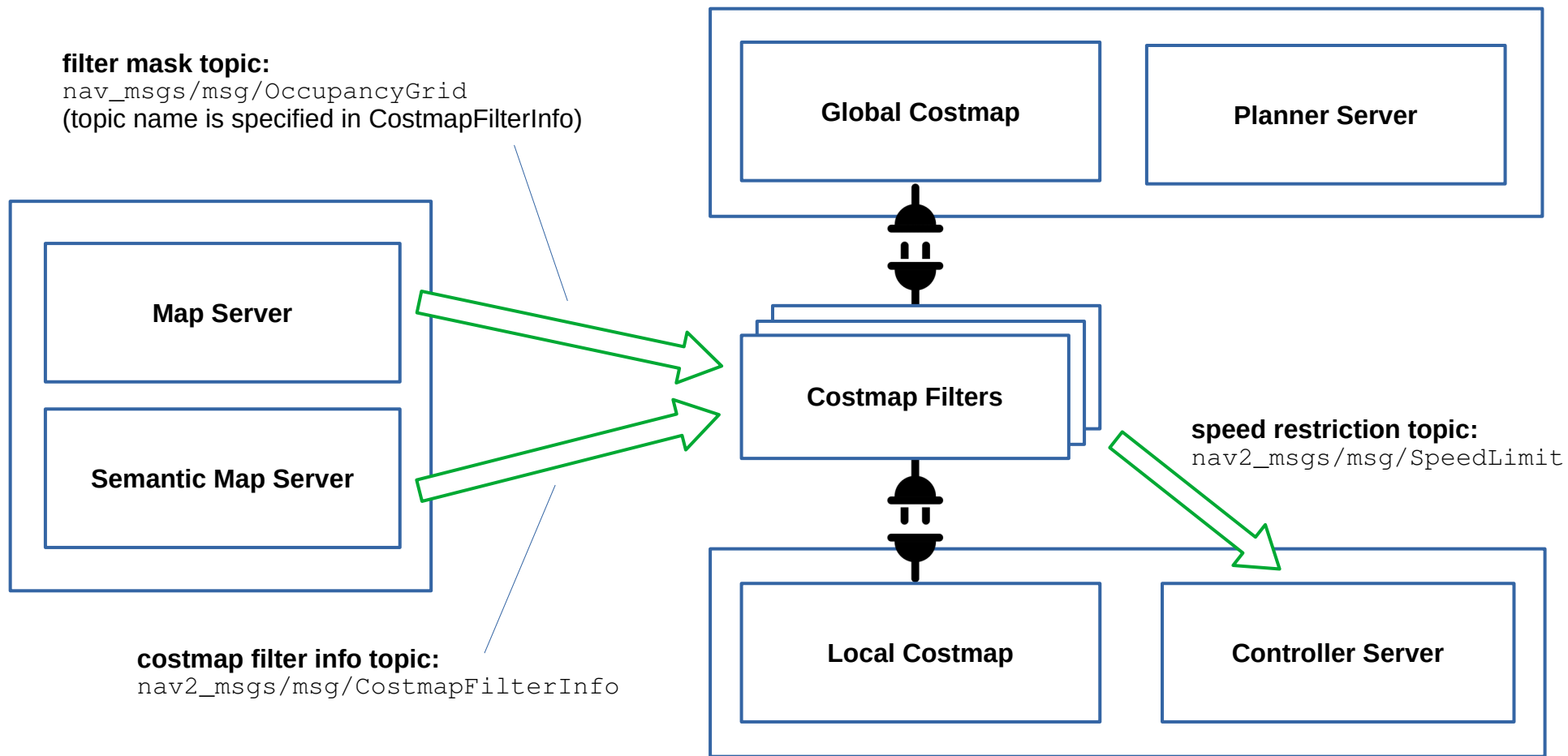


# Costmap Filters High-Level Design

v.1.1

## ROS2 Overall Structure



# Costmap Filters Structure

class  
inheritance

CostmapLayer

## CostmapFilter

```
class CostmapFilter {
public:
    virtual initializeFilter(
        const std::string & filter_info_topic) = 0;
    virtual process(
        Costmap2D & master_grid, int window_bounds,
        Pose2D & robot_pose) = 0;
    virtual resetFilter() = 0;

protected:
    pose last_robot_pose;
};

CostmapFilter::activate() {
    // Creates subscriptions to info and mask topics
    initializeFilter(filter_info_topic);
}

CostmapFilter::deactivate() {
    // Reset all subscriptions
    resetFilter(filter_info_topic);
}

CostmapFilter::reset() {
    ...
    // Reloads filter
    resetFilter();
    initializeFilter(filter_info_topic);
}

CostmapFilter::updateBounds(robot_pose, bounds) {
    ...
    last_robot_pose = robot_pose;
}

CostmapFilter::updateCosts(
    Costmap2D & master_grid, int window_bounds) {
    ...
    // An algorithm for how to use that map's
    // information. Fills the Costmap2D with
    // calculated data and makes an action based
    // on processed data.
    process(master_grid, window_bounds, last_robot_pose);
}
```

## KeepoutFilter

loadFilter() - Creates a subscription to filter\_info and filter\_mask topics.

process() - Based on loaded map composes Costmap2D with [FREE\_SPACE..LETHAL\_OBSTACLE] cost depending on data value in filter\_mask.

## SpeedFilter

loadFilter() - Creates a subscription to filter\_info and filter\_mask topics. Creates a publisher for speed\_limit topic.

process() - Checks whether the robot is entered or leaved marked on map speed restriction area. If entered checks the speed limit and sets speed\_limit topic value. If leaved restores back speed\_limit topic value to no-limit.

## LanesFilter

Covered by KeepoutFilter ([#1522](#)).

# Filter Topics

## **KeepoutFilter**

### Input topics:

filter\_info: nav2\_msgs/msg/CostmapFilterInfo

filter\_mask: nav\_msgs/msg/OccupancyGrid

## **SpeedFilter**

### Input topics:

filter\_info: nav2\_msgs/msg/CostmapFilterInfo

filter\_mask: nav\_msgs/msg/OccupancyGrid

### Output topics:

speed\_limit: nav2\_msgs/msg/SpeedLimit

## **LanesFilter**

*Covered by KeepoutFilter*

### **nav2\_msgs/msg/CostmapFilterInfo.msg:**

```
std_msgs/Header header
# Type of plugin used (keepout filter, speed
limit in m/s, speed limit in percent,
etc...)
uint8 type
# Name of filter mask topic
string filter_mask_topic
# Multiplier base offset and multiplier
coefficient for conversion of OccGrid data -
> into some other number space:
space = data * multiplier + base
float32 base
float32 multiplier
```

### **nav2\_msgs/msg/SpeedLimit.msg:**

```
std_msgs/Header header
# Setting speed limit in percentage if
true or in absolute values in false
case
bool percentage
# Maximum allowed speed (in percent of
maximum robot speed or in m/s
depending on "percentage" value). When
no-limit it is set to 0.0
float64 speed_limit
```