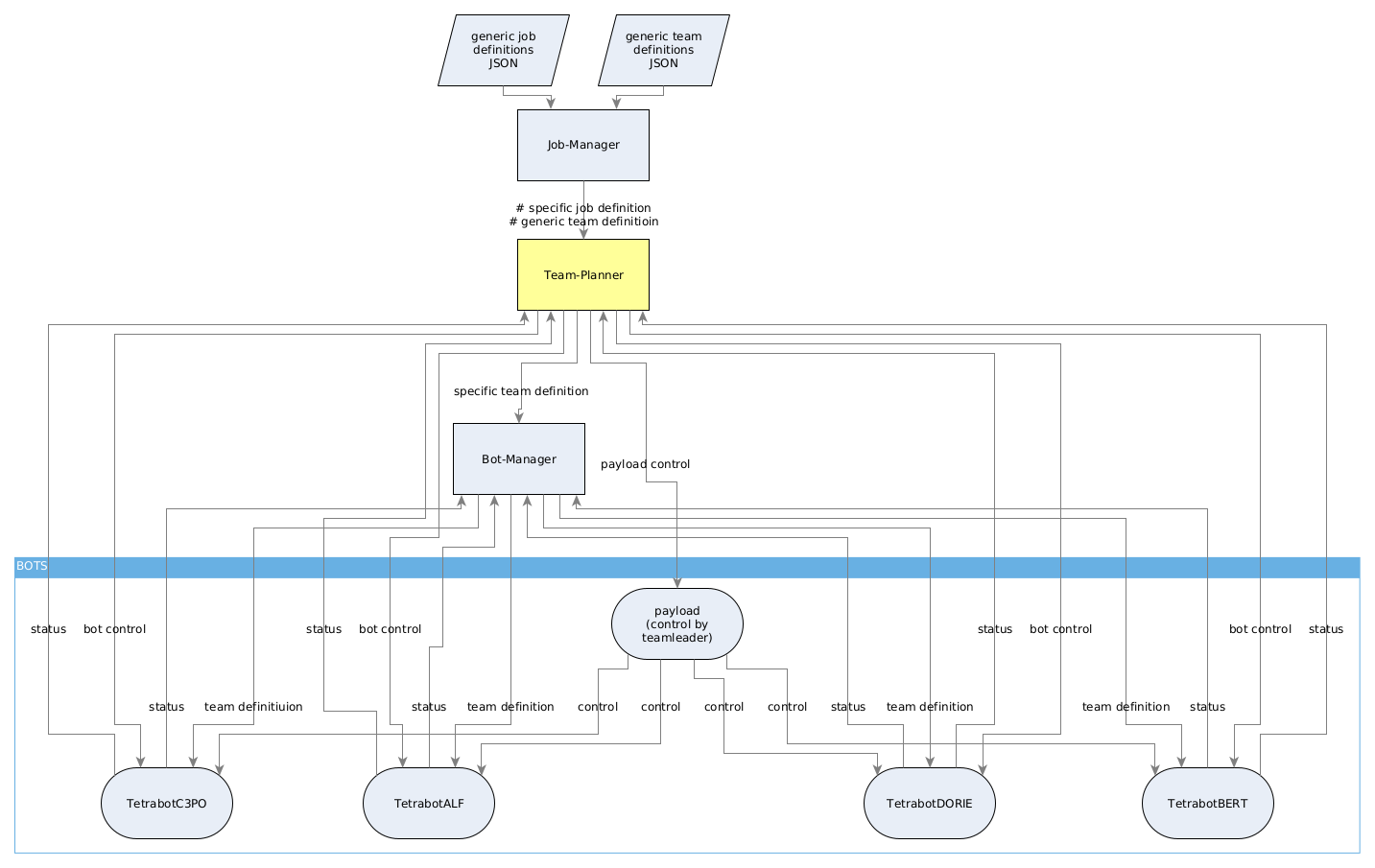
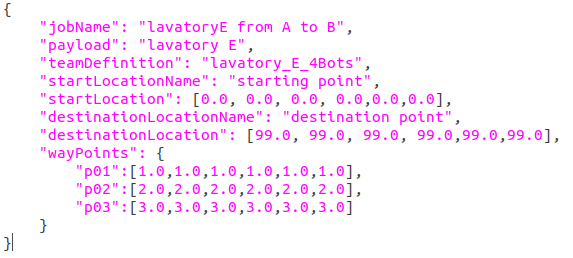
**Concept**

****

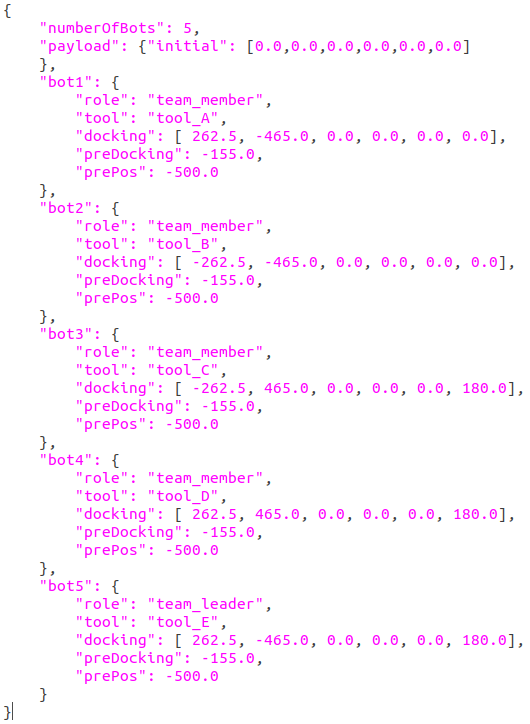
**GENERIC job-definition**

The generic job-definition contains basic information about a job, like payload, start location, destination location and way-points. In addition it contains the name of the generic team definition.



**GENERIC team-definition**

The generic team-definition is a geometric definition for a specific payload. It contains the number of bots needed, docking- pre-docking and pre-position for each bot in relation to payload center, the required mechanical interface per bot and the role-definition (leader or member).

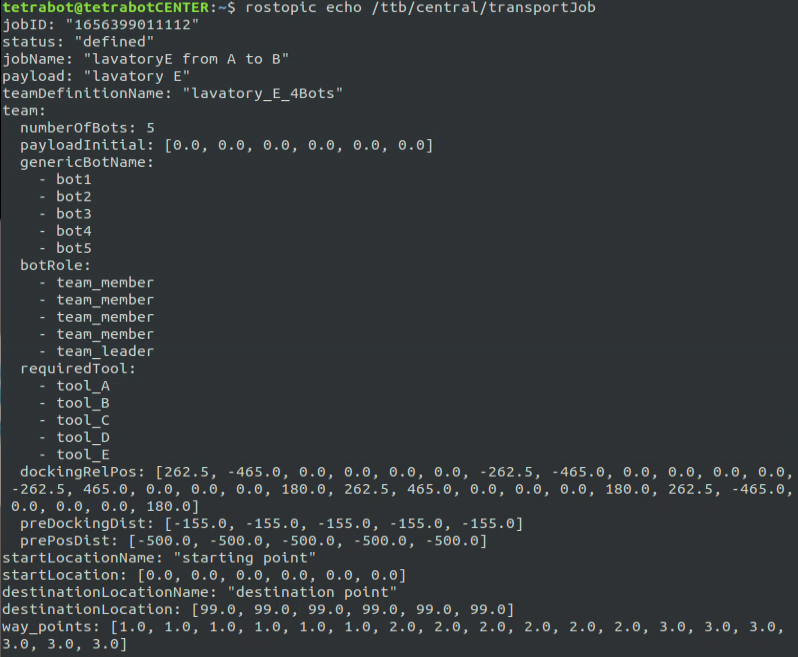


**JOB-MANAGER**

The Job-Manager starts a new transportation-job by selection of the pre-defined generic job-definition. The Job-Manager integrates the generic Team-Definition specified in the generic Job-Definiitoin and combines both to a Job-Message, published as:

**/ttb/central/transportJob**

**example:**

****

**TEAM-PLANNER**

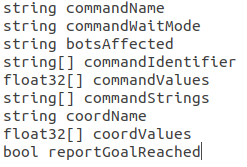
The team-planner combines the generic job-information, the generic team information and the status information of each bot to build the team. This specific team information has to be handed over to the bot-manager (how? tbd), to the bots (how? tbd) and it has to change the transport-job status (how? tbd).

**teamCommand.msg**

The teamCommand.msg is used to send commands to specific bot(s) (one or several) or to control an entire team.

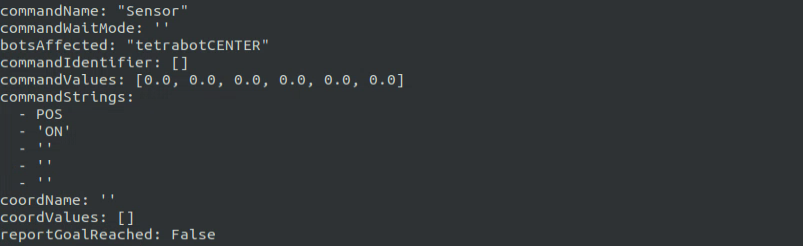
For specific bot control the teamCommand.msg is published to **/ttb/tetrabotCENTER/ext/teamCommand,**

For entire payload conrol it is published to **/ttb/tetrabotCENTER/ext/payloadCommand**

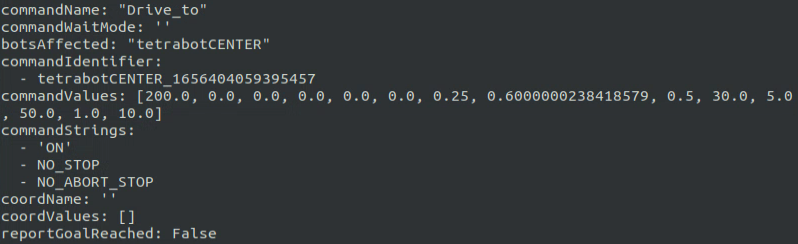


The teamCommand is a container for different commands and it may contain different parameters. Main content is the “commandName”, the “botsAffected”, the “commandValues” (floats) and the “commandStrings” (strings).

example to turn position signal ON:



example to drive to specific position



**sample script**

Commands can be arranged in so called scripts

