

Hallo das ist ein Test.

- **Packages:**

Die Softwareprojekte werden in ROS in *Packages* verwaltet und organisiert. Ein Paket enthält alle Laufzeitprozesse eines Projektes, die Programmbibliothek, Datensets, Konfigurationsdateien und gegebenenfalls Softwarekomponenten von Drittanbietern. Die Verwaltung eines Projektes in Paketen ermöglicht eine einfache Modularität der Funktionalitäten.

- **Nodes:**

Nodes stellen in ROS ausführbare Prozesse dar. Diese Instanzen sind verteilt über die Kommunikationsinfrastruktur, können aber durch verschiedene Mechanismen zur Laufzeit miteinander kommunizieren oder Daten austauschen. Typischerweise werden die thematisch entkoppelten Funktionen eines Projektes in einzelne Knoten verteilt.

- **Topic:**

Topics sind Datenbussysteme, über die zur Laufzeit Nachrichten in asynchroner Form ausgetauscht werden. Dazu verwendet ROS eine *Publisher/Subscriber*-Semantik. Die Knoten publizieren dabei Informationen auf verschiedene Topics, während andere Knoten diese Topics beliebig abonnieren können, um die Informationen auf dem Datenbus einzulesen. Diese Funktionsweise ermöglicht es, einzelne Softwarekomponenten unabhängig entwickeln und in bestehende Projekte integrieren zu können.

- **Messages:**

Die Informationen auf den Datenbussen der Topics werden in Form von *Messages* in standardisierten Datenstrukturen versendet. Es existiert eine Vielzahl von nativen Datenformaten einzelner Botschaften (boolean, integer, floating point, usw.), es lassen sich aber auch eigene Messages aus Arrays der definierten Datenformate generieren.

- **Services:**

Gegenüber dem Broadcast-Prinzip der Subscriber und Publisher lassen sich mit *Services* in ROS Informationen auch auf explizite Requests erhalten. Dabei wird von einem Knoten eine definierte Informationen durch einen Service von einem einzelnen Knoten angefordert. Der Client-Node antwortet dann auf diese Anfrage mit dem Senden der entsprechenden Message an den anderen Knoten.