

- *Keyword (Wake-Word, Hot-Word, Keyphrase atd.)* – specifické slovo nebo fráze, která po vyslovení může aktivovat neaktivní zařízení, například: "Hey Siri", "OK Google" a tak dále.

2.3 Framework ROS

Robot Operating System³ (ROS) je open source framework spravovaný společností Open Robotics se sadou softwarových knihoven a nástrojů pro vytváření robotických aplikací. ROS poskytuje hardwarovou abstrakci, nízkouúrovňové ovládání zařízení, předávání zpráv mezi procesy, nabízí ovladače zařízení, knihovny, vizualizéry, správu balíčků a další funkce. Tento framework implementuje několik různých stylů komunikace, včetně RPC synchronní komunikace realizovatelnou přes služby (Services), asynchronního streamování dat přes témata (Topics) a ukládání dat na serveru (Parameter Server) používaného uzly (Nodes) za procesním běhu [11]. Mezi základní prvky ROS architektury patří:

- *Package (Balík)* – hlavní jednotka pro organizaci softwaru v ROS. Slouží jako strukturovaný kontejner pro organizaci a správu softwarových komponent nezbytných pro efektivní strukturování, sdílení kódu, dat a zdrojů v rámci ekosystému ROS [12].
- *Node (Uzel)* – nezávislý modul, který v softwarové architektuře zařízení vykonává určitou funkci nebo úkol. Tyto moduly mohou vzájemně komunikovat, aby dosáhly celkové funkcionality. ROS systém se obvykle skládá z velkého množství uzlů [12].
- *Message (Zpráva)* – jednoduchá datová struktura sestávající z typizovaných polí obsahujících standardní primitivní typy (integer, float, boolean, atd.). Uzly komunikují mezi sebou pomocí zpráv [12].
- *Topic (Téma)* – speciální kanál, prostřednictvím čehož uzly mohou publikovat a přijímat zprávy. Je to prostředek pro výměnu dat mezi uzly, který jim umožňuje vzájemnou interakci a spolupráci v ROS systému [12].
- *Services (Služby)* – speciální typ komunikace využívající témata, který umožňuje uzlům provádět vzdálené operace a požadavky. Služby jsou založeny na modelu požadavek-odpověď. Jeden uzel odešle požadavek (request) na provedení určité operace a jiný uzel tento požadavek splní a odešle zpět odpověď (reply) [12].

³<https://www.ros.org/>

- *Actions (Akce)* – komunikační model pro provedení dlouhotrvajících vzdálených příkazů a požadavků, má podobný typ komunikace jako model *Services*. Tento typ komunikace umožňuje sledovat průběh požadavků, získat konečný výsledek a v případě potřeby požadavek zrušit před jeho dokončením. Pro tyto všechny účely slouží balíček *actionlib* [13].

Lze také zdůraznit hlavní cíle a úspěchy ROS, které jsou popsány na oficiálních webových stránkách [11], jmenovitě:

- *Jazyková nezávislost*: framework ROS lze snadno implementovat do jakéhokoli moderního programovacího jazyka, například: Python, C++, Java atd.
- *Snadné testování*: ROS má vestavěný framework pro jednotkové/integrační testování s názvem *rostop*, který usnadňuje vytváření a odstraňování testovacích přípravků.
- *Škálování*: ROS je vhodný pro velké běhové systémy a pro velké vývojové procesy.

2.4 Tkinter

V určitých situacích může být velmi obtížné stanovit správnost výpočtů nebo popsat princip fungování zařízení pomocí pouze čistých dat. Aby bylo snazší identifikovat chyby nebo vysvětlit některé z jevů, se vytváří vizuální model, kopírující skutečné chování určitého zařízení. Pro tyto účely v bakalářské práci byl zvolen Tkinter⁴ framework. Tkinter je vestavěný modul jazyka Python, který se používá k vytváření aplikací s grafickým uživatelským rozhraním (GUI). Framework je open-source vydán pod Python licenci. Tento nástroj má jednoduchou strukturu vytváření vizuálních prvků a snadno se s ním pracuje, proto Tkinter je jeden z nejčastěji používaných modulů pro vytváření GUI aplikací v Pythonu. Rozhraní Tkinter frameworku je založeno na multiplatformní sadě nástrojů Tk⁵, která byla původně navržena pro Tool Command Language⁶ (Tcl) [14, 15].

- *Graphical User Interface (GUI)* – forma uživatelského rozhraní, která umožňuje uživatelům komunikovat s počítačem prostřednictvím vizuálních ukazatelů (ikony, okna, tlačítka atd.).

⁴<https://docs.python.org/3/library/tkinter.html>

⁵[https://en.wikipedia.org/wiki/Tk_\(software\)](https://en.wikipedia.org/wiki/Tk_(software))

⁶<https://www.tcl.tk/>

Widgety jsou hlavními prvky GUI aplikace ve frameworku. Pomocí widgetů, které jsou součástí sady nástrojů Tk, Tkinter poskytuje uživatelům jednoduchý způsob vytváření elementů grafického uživatelského rozhraní. V aplikaci lze widgety Tk použít ke konstrukci tlačítek, textových boxů, datových polí atd. Po vytvoření tyto prvky lze propojit nebo interagovat s funkcemi, metodami, datovými proměnnými nebo dokonce s jinými widgety [14, 15].

Vlastní metody a jejich aplikace

3

- 3.1 Robotická entita**
- 3.2 Detekce směru a polohy řečníka v prostoru**
- 3.3 Grafická vizualizace a modelování chování cílového zařízení**
- 3.4 Implementace technologie KWS**