ERICA

Empathic Robotic Interactive Companion and Assistant – QC Enterprises

Business Case

At the heart of the QC product line lies ERICA, a general purpose robot for the elderly, capable of making coffee and monitoring client's health. A social and lovable robot, she is also customizable in appearance and extendible in capacity. As the first entry in the QC Robotic Universe, she is a jack of all trades, but master of none and will thrive if supplemented by more specialized QC robots. By entertaining clients, who are often lacking (or missing frequent) visits of family or friends, ERICA improves quality of life.

Scope

During the conceptual phase of ERICA's design we touched on the topic of a supporting infrastructure. Any robot, let alone ERICA, is a highly complex device and will likely require frequent updates preferably pushed over the internet. In addition, the modular functionality approach of ERICA demands a marketplace, possibly integrating third-party solutions. While this platform is interesting, for this project we'll assume this platform to a) exist and b) be sufficient for our international ambitions.

Functionalities

ERICA has some functionalities which are briefly described below. The system

1. Integrates with QC Online, a platform which facilitates remote monitoring, allows push-updates, integrates customer support and troubleshooting, gathers intelligence on usability and target audience and permits sale of additional modules and subscription based content
2. Performs day-to-day tasks of a personal assistant. This can include opening the door, making coffee, and fetching objects that would otherwise require standing up and walking.
3. Actively engages in social activities and facilitates (remote) communication. Examples of simple social activities can be playing fetch or card games with the user. ERICA also engages in social activities with care-bots or pets of other users. This encourages the users to interact with each other, lowering the threshold for social interaction. Facilitation of communication includes an interface for video calls, and the ability to assist in conversation if the user has trouble speaking or hearing. The ERICA system is also able to detect medical and other emergency situations and contact the authorities or a general practitioner.
4. Has an AI with simple personality features and a customizable appearance. Personality features are behavior patterns that suit the user, such as the playfulness or social activeness of the robot. The appearance can be customized in different ways. For this AI the assumption has been made that this is achievable in the near future.

Stakeholders

Stakeholders within QC Enterprises

Many entities have interests in ERICA. First of all, these are entities within the developing company, QC Enterprises. The shareholders are: the software architecture team, the development team, the maintenance team, the customer support department and the marketing department. The shareholders have interest in ERICA, because selling it can make money for QC Enterprises, but the development of ERICA will cost the company money.

Secondly, the software architecture team has interests in the system, because it must develop the architecture for ERICA, and the company's update platform must be able to release updates for ERICA. Furthermore, the development and maintenance teams have interest in ERICA, because they will need to make adaptations to its software, which therefore needs to be easily maintainable. Another stakeholder of ERICA within QC Enterprises is the customer support department, which needs to be able to support users in the use of ERICA and thus needs the system to be as easy to use as possible. Finally, the marketing department needs to be able to sell ERICA to customers.

External stakeholders

Besides stakeholders from within QC Enterprises, other entities have interests in ERICA too. These are he users of the system. Above all, the elderly who will have ERICA in their houses have interests in the system. They need ERICA to behave as expected, safely and within reasonable time, after telling the system what to do with as less effort as possible. People visiting the houses of other people using ERICA have interests in the system too, because they would not like to be disturbed by the presence of ERICA. In the case of family, friends and home care providers, they also want ERICA to help its owner as good as possible. Lastly, healthcare providers of users of ERICA can get information about the lifestyle of their clients or patients from the system.

Future stakeholders

In the future, ERICA might be extended with medical functionalities. When this is done, ERICA must comply to medical regulatory, and in a sense, medical regulatory agencies become stakeholders. This will add new requirements on the subject of safety of use and certification by them will be necessary to sell ERICA with these functionalities added. These extra requirements are the reason a first version of ERICA will not have any medical functionalities. When these functionalities are added, insurance companies will become stakeholders too, because they will have to pay for ERICA when it has medical functionalities.

Overview over all stakeholders

The table below summarizes all stakeholders mentioned above and their interests in ERICA.

Stakeholder	**Current/future**	**Interests**
Shareholders	Current	Stakeholders invest money in the company and product, this is why they have interest in ERICA becoming a success and getting a return on investment.
Software architecture team	Current	Developing ERICA's architecture Extend update-platform for ERICA
Development and maintenance team	Current	The development and maintenance team have the interest to keep the system easily maintainable, this way if there are updates to be made to the system, these can be created without much problems. They would also want the software to be easily understandable so that new developers can understand the system without much effort.

Customer support department	Current	To help in the process of supporting the customers the customer support department has interests in an easy to use and understand system. This way it is easier for them to understand the system and help the customers with their problems and their solutions.
Marketing department	Current	To make ERICA a success, the marketing department has interest in ERICA being as easy to sell as possible. For instance if it's not to expensive, easy to use.
Elderly	Current	Because most elderly are not very technical, ERICA has to be easy to use for them. Otherwise they will not get all the functionality out of it. Because elders are vulnerable the system has to be very safe and not harm them in any way. Effectively this means the robot may not run into the elderly or harm him / her with its arms. The system also has to be fast in aiding the elder, otherwise might find it annoying.
Visitors of people using ERICA	Current	People unfamiliar and familiar with ERICA should not be hindered in any form or way by it. ERICA only has a helping role in these situations. Visitors can be interested in ERICA helping as a translator, repeating sentences if the user did not hear it correctly.
Family, friends and home care providers of people using ERICA	Current	Friend, family and home care providers cannot be around all the time, for this reason they have interest in ERICA for it can watch the person 24 hours a day, 7 days a week. This takes a huge burden from them. And also they don't have to worry about their loved one / patient.
Healthcare providers of people using ERICA	Current	The healthcare providers have interest in ERICA as with the activity monitor, they can track the client's / patient's health. This way they can react to an unhealthy lifestyle, but don't have to be

		around the client / patient all the time.
Medical regulatory agencies	Future	When medical functionalities are added to ERICA, it will have to comply to all sorts of new requirements, for this reason they will become stakeholders in the future.
Insurance companies	Future	If we are going to sell the product through the insurance companies, they would want to spend as little as possible, but they benefit from ERICA in the way that ERICA improves the quality of life of the user, and positively impacts the lifestyle.

Requirements

Mobile assistant, autonomous aid for day-to-day tasks of the elderly

ERICA must be capable of performing common day-to-day elderly tasks, like making coffee and opening doors. For this reason it must be mobile and needs to be capable of reaching all places in an elderly home. It must also be mobile in the sense that for most of the day it is available for use, i.e. not restricted by battery. Stairs are of no concern to ERICA, as most elderly homes do not have them.

Modular, functionality must be expandable through (purchase of) additional modules

There is no consensus on what makes a household robot 'good' and due to the small size and low-budget of QC Enterprise, developing all functionality before release will be impossible. Therefore it is critical to make ERICA modular, permitting gradual development and deployment after release. Considering the expected success of ERICA, purchase of these modules must be delivered through an online platform, though offline options are a future option. Updates and third-party services connected to the platform also aid maintenance.

Lovable and social, AI must have a personality

The elderly are mostly lonely and receive too little attention from their families and loved ones. The huge success of Paro in Japan signals the need for a certain cuteness in both the appearance and character of ERICA. She must have personality traits, adapt to client behavior and allow engage in conversation. Usability has a high priority, and ERICA must not contain

Customizable, through offline and DLC purchase of hats

People like to differentiate on objects they consider important in their lives as seen in fashion, cars and pets. Cats and dogs are different by nature, but ERICA should reflect this customization aspect. Considering ERICA will likely be found in high concentrations, as most lonely elderly live in elderly homes, this need to feel unique is enlarged.

Responsible, capable of monitoring client and alerting authorities when in need

When alone, an elderly person tripping over a wire or falling down stairs is often fatal. ERICA can fulfill the role of an always present guardian and must be able to notify emergency services when in need. Basic medical monitoring is also required, which may in future be coordinated with a general practitioner.

Safe and marketable

Elderly and the people visiting them are often vulnerable. Therefore, ERICA needs to perform it's tasks without causing danger to people around the system.

Reference Architectures

There have been multiple medical and helping robots in the past, this chapter will describe some of their architectural decisions which are of interests of us.

HERB

HERB is a Home Exploring Robotic Butler built by Intel. The robot, built on top of a Segway base, has a robotic arm and several sensors. The system's architecture was designed around two driving principles: the availability of unlimited computational power and minimal human input into sensing and planning algorithms. The architectural decision of interest for the development of ERICA is to offshore most of the computations. This keeps the power consumption of the robot as low as possible and thus increasing the battery life. This also gives the developers more computing power for things such as object recognition.[1]

GiraffPlus

GiraffPlus is a complex system which can monitor activities in the home. This is done with sensors in the home and on the body. The center of this system is a telepresence robot, which allows elders to communicate with relatives and health care professionals. This social aspect has been of interest for us from the very beginning. The telepresence robot itself is not capable of much more than just facilitating social interactions, but can be a good addition to the ERICA platform. The monitoring system on the other hand can detect several things in the home. For instance it is able to detect blood pressure and if the elder has fallen, and then contact the appropriate person or professional.[2]

[1] https://www.ri.cmu.edu/pub_files/2010/1/HERB09.pdf

[2] http://www.giraffplus.eu

PARO

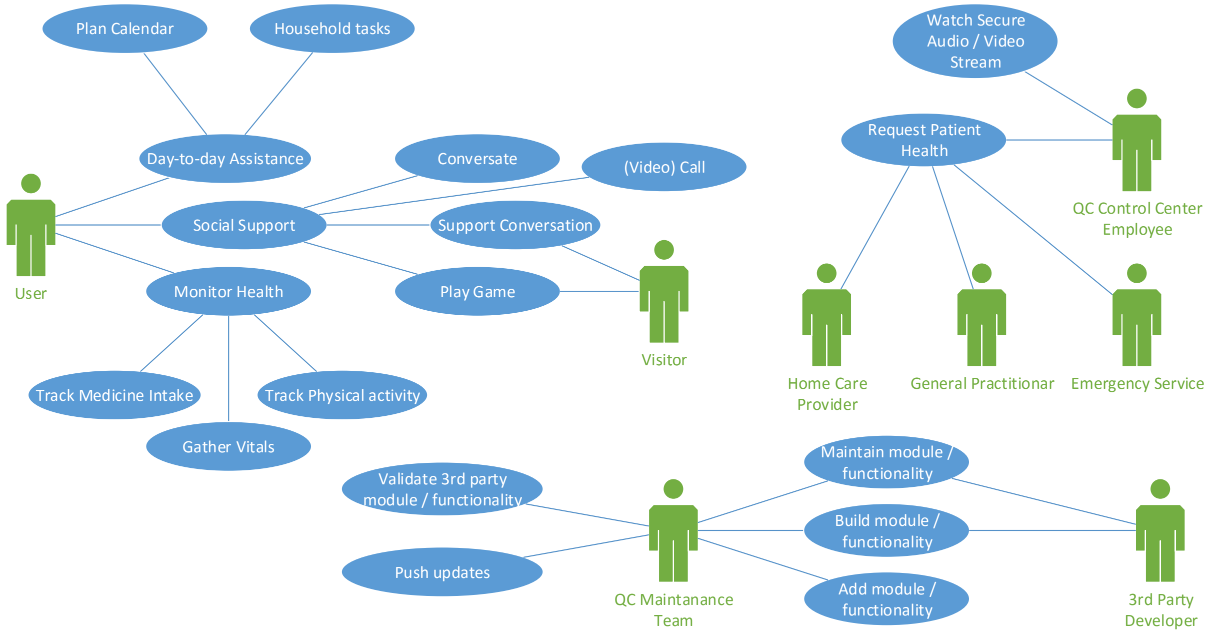
PARO is an advanced therapeutic robot, which allows for animal therapy in environments which do not allow for animals. It improves the relaxation and motivation of patients, helps reduce stress and improves the interaction between patients and caregivers. The patient can also influence the behavior of PARO by stimulating or discouraging the behavior as it happens. This learning and motivating is something which might also be implemented in ERICA, to make the user feel at ease.[3]

[3] http://www.parorobots.com

Architecture overview

The 4+1 view model will be used to describe the architecture of ERICA.

Use case

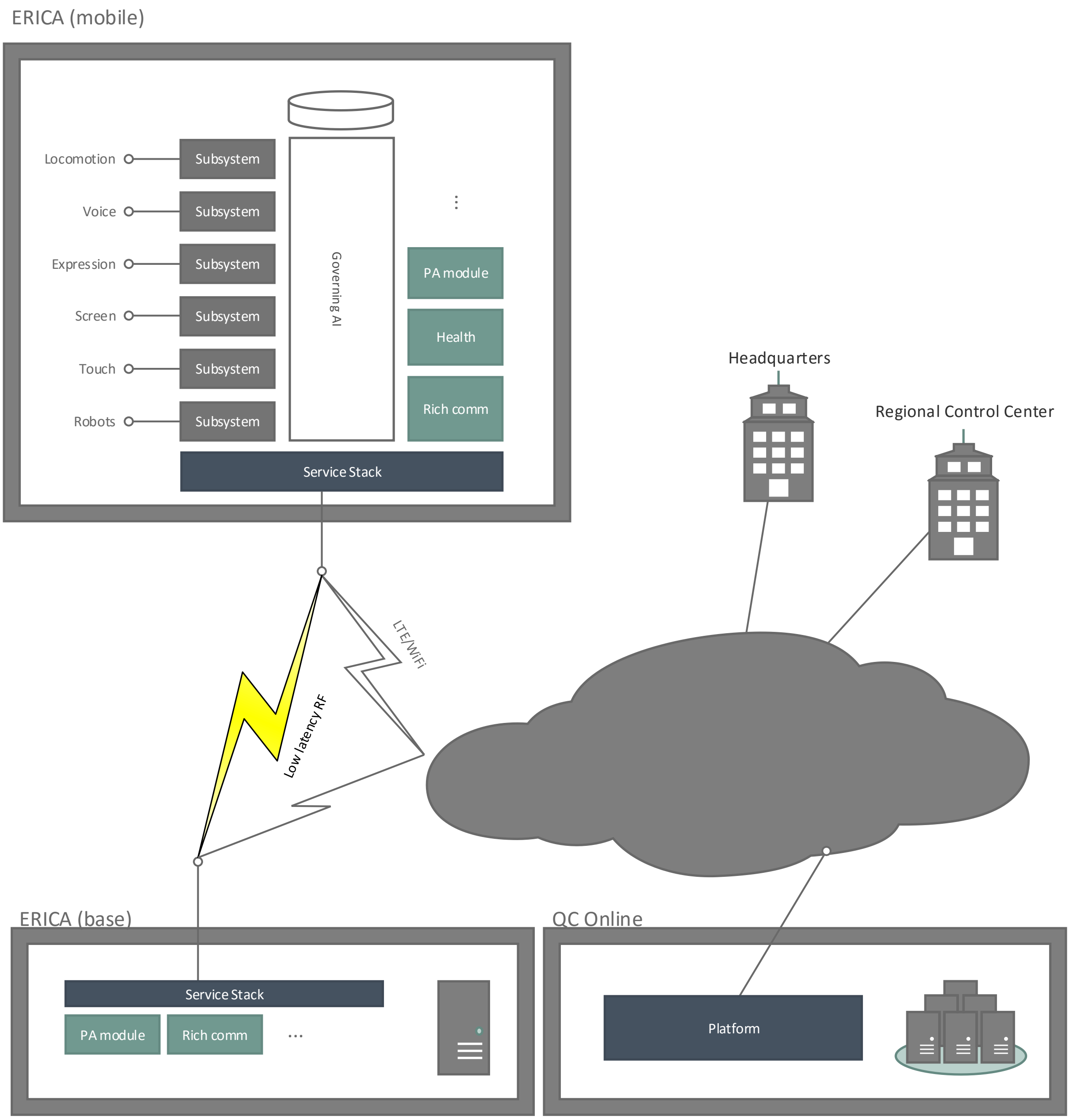


The above diagram documents the actors in the ERICA system and their desired functionality. From the user's perspective ERICA should be a highly functional robot, capable of socializing and easing day to day tasks, as well as functioning as a health monitor. Communication of these medical insights is the primary concern for care providers and GP's, though emergency services should be able to access the data if possible. Development is primarily concerned with modular development and the ability to push updates. Third party modules can be added to ERICA, though these must always to through extensive validation by QC staff.

Logical view

The ERICA system is the combination of three physically separate systems, namely:

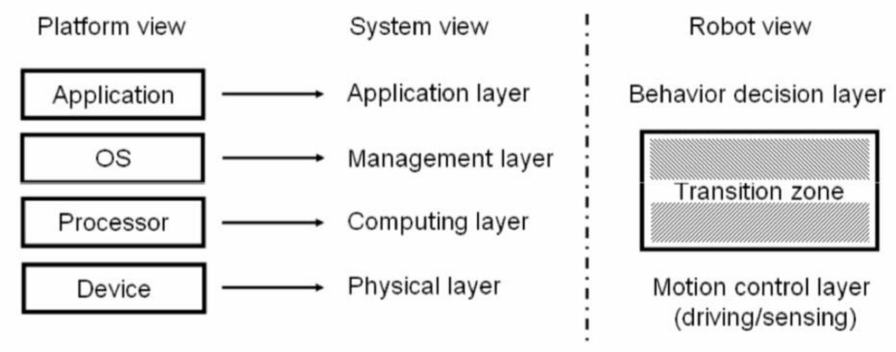
The robot is the main actor and holds most hardware capable of interacting with the environment. Due to the energy and space constraints imposed by the robot's mobility, more powerful computing is available from a local base station. Finally, synchronization, data services and insights are communicated through the QC Online platform.



Presented above is the general logical view for ERICA. Components in the architecture are:

1. **Off-the-shelf hardware subsystems**, responsible for vision, output and environmental actions;
2. **Governing AI**, with persistent personality and capacity to reason about available resources;
3. **Functionality modules**, providing integration services and high-level functionality;
4. **Service stack**, providing a consistent mechanism for updates, communication and diagnostics;
5. **Platform**, delivering data services, backup functionality and intelligence.

Process view



Central to the implementation and process views is a set of middleware called Robot Operating System, or ROS[4]. It offers an integrated approach to robotic development and focuses on peer-to-peer (i.e. distributed) computing, multi-language and reusable components. Due to the *heterogeneous* nature of robotics is key to both ROS and ERICA's architecture. Rarely is a system as easily consolidated as a desktop application. ROS offers a way to spread computing across heterogeneous systems through *nodes*. It also provides standard operating services such as hardware abstraction, low-level device control and package management.

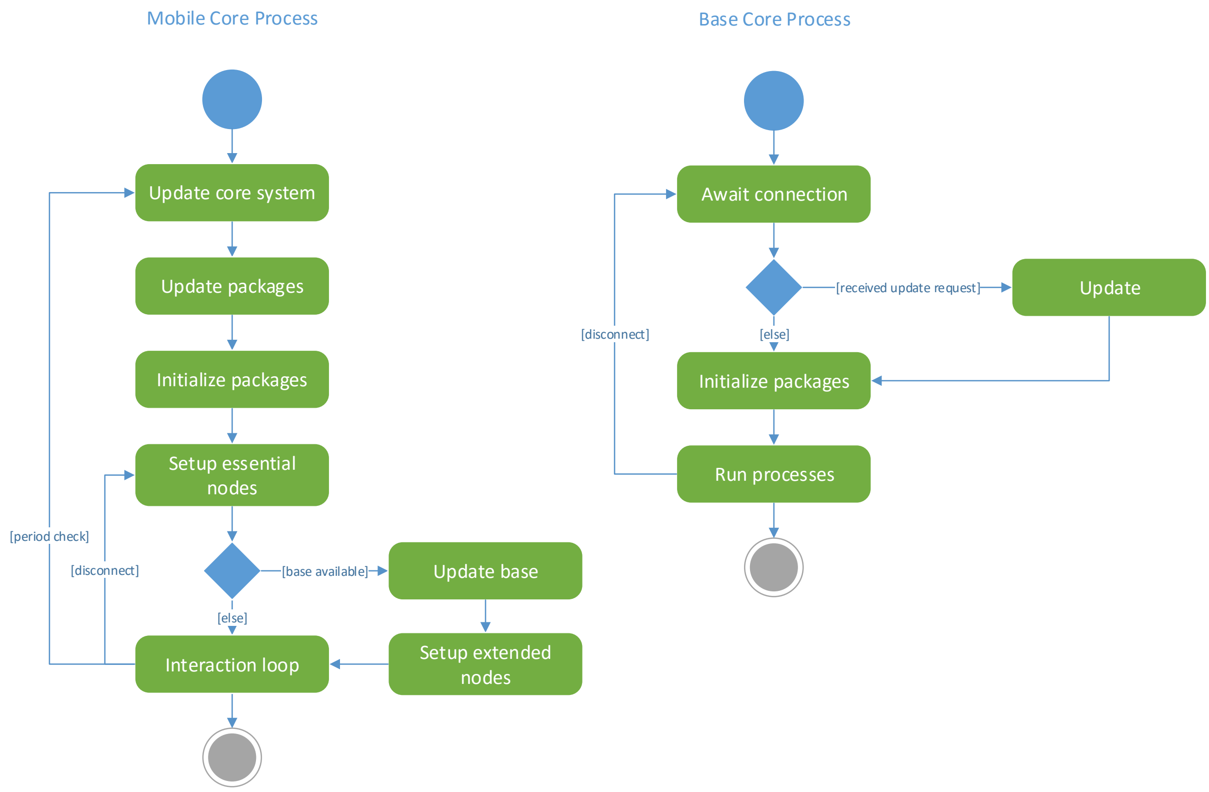
Software in ROS is organized in packages. A package might contain nodes, a ROS-independent library, a dataset, configuration files, a third-party piece of software, or anything else that logically constitutes a useful module. The goal of these packages it to provide this useful functionality in an easy-to-consume manner so that software can be easily reused. This reusability means ERICA is able to use off-the-shelf systems for its peripheral features (vision, locomotion) and also provides a way to integrate and deploy custom components.

ROS requires a *coordination node*, responsible for the creation and deployment of other logical nodes including the capability to *offshore* computing to a remote system. As the wireless nature of the robot means a connection to the base station cannot be guaranteed, and it cannot be expected of an autonomous robot to be fully dependent on such a system, the mobile system is best suited for this role.

By embracing ROS's organizational structure processes are easily structured as crucial aspects such as inter-process communication and distributed data storage can be tackled with relative ease[5]. In terms of processes, ERICA is modular to a large extend. At the time of writing, no details are available on specific subsystem requirements, though it is known that both mobile and base systems run at least one process responsible for coordinating each system. These processes may spawn or startup new processes depending on hardware and role. On the next page the lifecycle of both processes is shown.

[4] http://www.generationrobots.com/en/content/55-ros-robot-operating-system

[5] QC online will run from a PaaS solution such as Azure or AWS ensuring scalability, though further details will be omitted.



The lifecycles described above are mainly concerned with delivering updates and initializing ROS nodes. These processes are different on mobile and base, as the base system serves as an *extension* of the mobile unit. Modules running on the base are likely highly reliant on those running on the mobile unit, therefore the mobile unit is responsible for determining the software present on the base system.

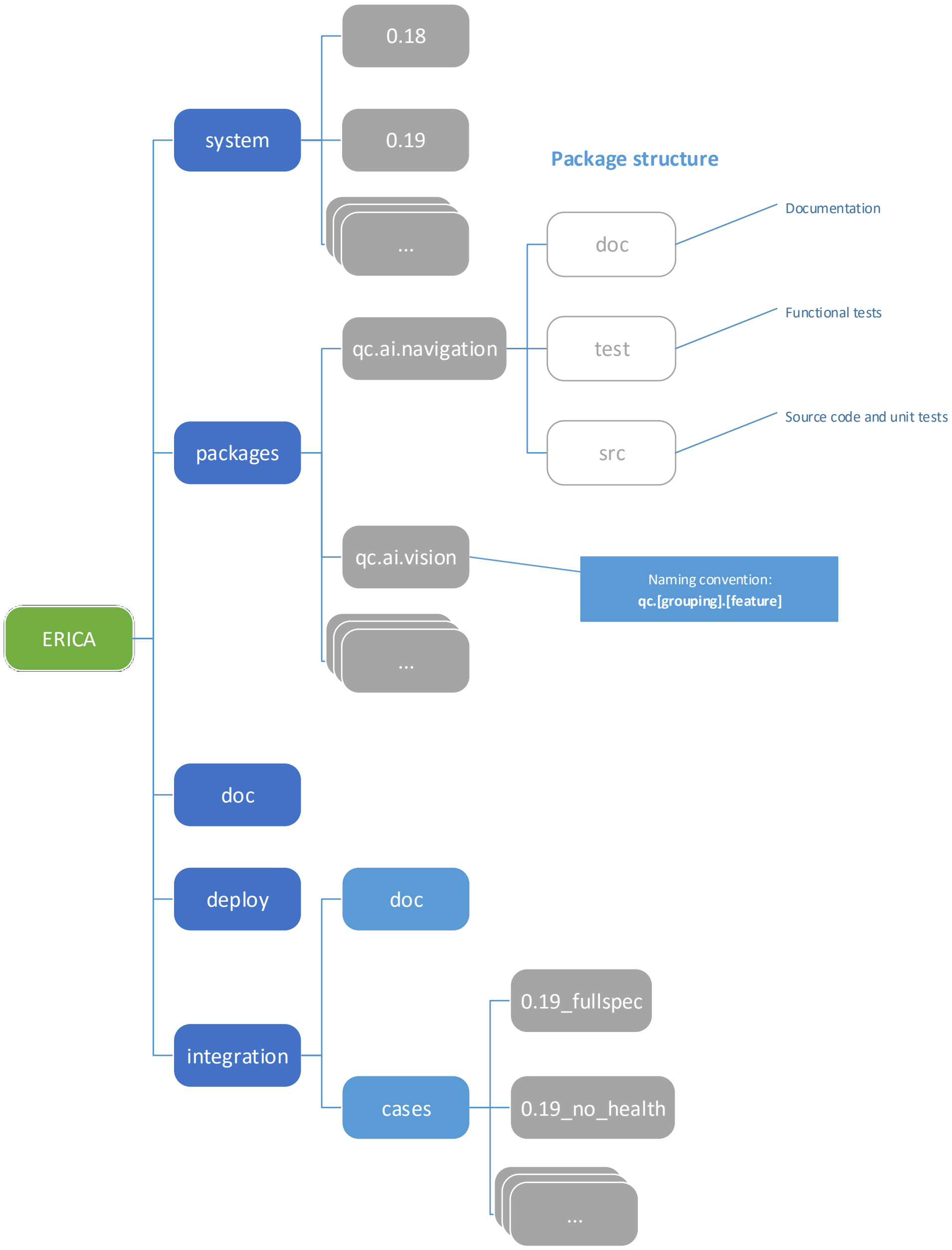
Under 'core' system, everything down from the management layer shown on the previous page is understood. This includes Unix OS updates and firmware upgrades crucial to the functioning of the core process. Subsystems are updated through packages, as they likely require highly specific or vendor dependent code. All package management is performed by QC Online, responsible for maintaining an archive of versioned packages and system updates. Should the connection to the base system be lost resulting in another node setup, only packages dependent on the base node should be affected. This means the core AI process will not be killed in such an event.

Nodes are connected through network spanning multiple physical systems. The physical systems are connected through a *to-be-decided* high speed wireless interconnect which delegates connectivity to hardware drivers and the ROS. Internal systems are connected through either a high-speed Ethernet network or a custom interconnect, depending on latency requirements and hardware availability.

Implementation view

Most of ERICA's functionality will be delivered through ROS packages, the remainder being system updates or QC online related development. A ROS package might contain ROS nodes, a ROS-independent library, a dataset, configuration files, a third-party piece of software, or anything else that logically constitutes a useful module. They can also be grouped into *meta-packages*, or groups of packages. These packages can define *dependencies*, allowing modular development and testing. Occasionally, system-wide updates may be necessary which may involve kernel updates, key firmware and driver updates and common libraries. These are pushed through the UNIX distribution system.

Most development will focus around the creation, deployment and testing of ROS (meta-) packages which requires a strict source control and naming regime. An outline of the ERICA's source repository is given below.



Package development

- Packages may be developed in any language suitable for their purpose (i.e. C for drivers, Python for AI interaction, Java for web integration)
- Packages must have an *identifier*, following the naming convention *qc.[grouping].[feature]*
- If the package uses a language with namespace support, this identifier must be the namespace of the program
- Packages **must** contain three directories:
    - /doc/ for documentation
    - /test/ for functional tests
    - /src/ for source code and unit tests
- Packages are versioned *in whole integer increments* (1, 2, 3, …) and must be tagged in Git
- The *src* directory of each package must contain an XML file (*package.xml*[6]) or must produce one on build

Integration test development & automated testing

- For each deliverable product, an environment with specific test cases must be conditioned
    - Deliverable here means 'ERICA + feature set'
- Packages may at any point be pushed to the build server, which will run a full or partial test suite
- Once tagged, a package will automatically be pushed to the build server
- Before pushing to QA, a package must be tagged and build without warning

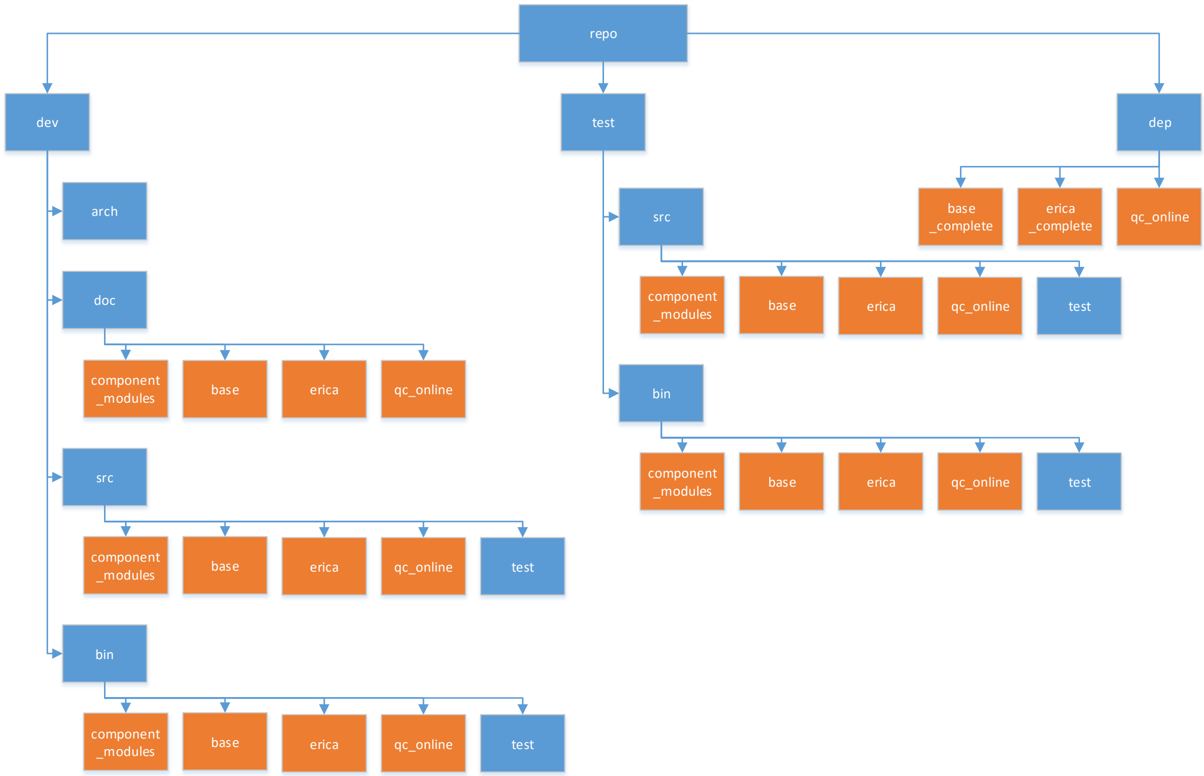
Examples of packages

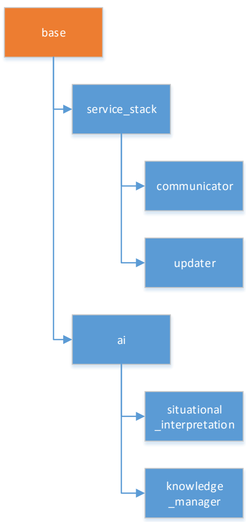
- Service
    - Updater, integrates package management and system updates with QC online
    - Sync, synchronizes local data storage with QC online
- Hardware interfaces
    - Vision, hearing, speech, movement, etc.
- AI components: task scheduling, navigator, situational awareness
- Functionality modules: personal assistant, health, social

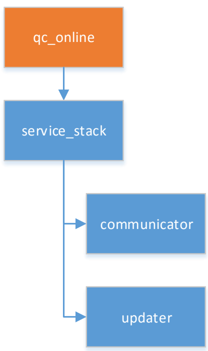
[6] See http://wiki.ros.org/catkin/package.xml for specifications

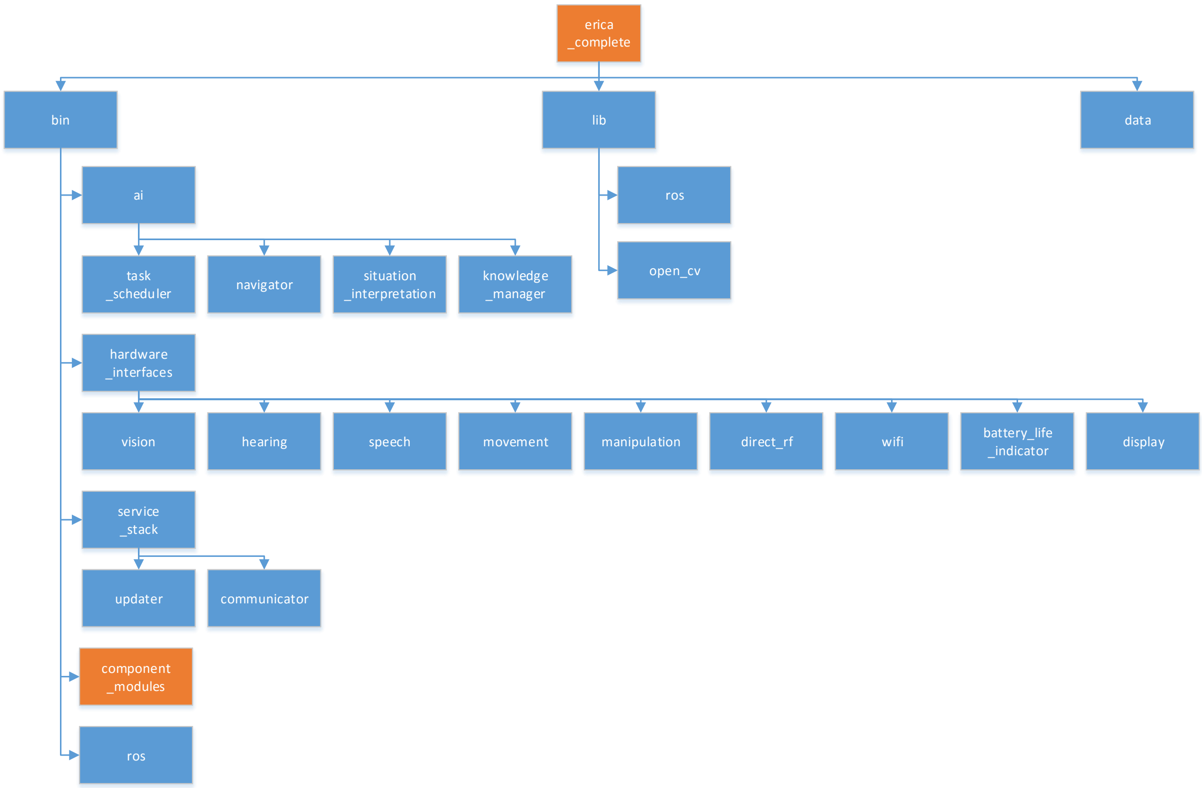
Prototype directory structure

The following hierarchies define the directory structure which might be used for the ERICA prototype.



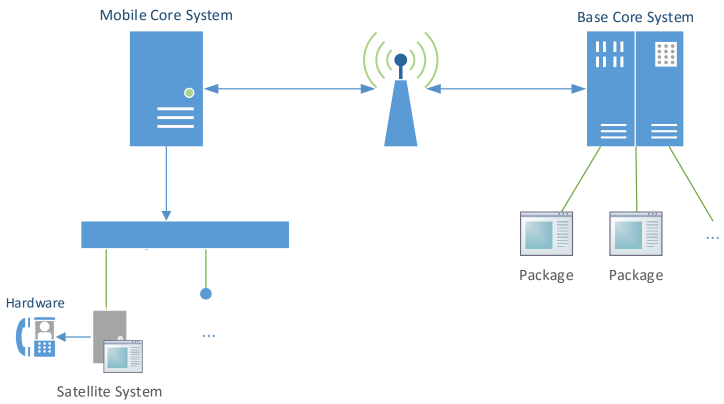




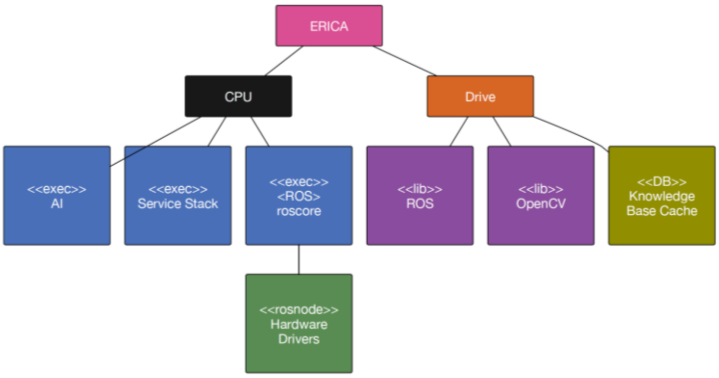


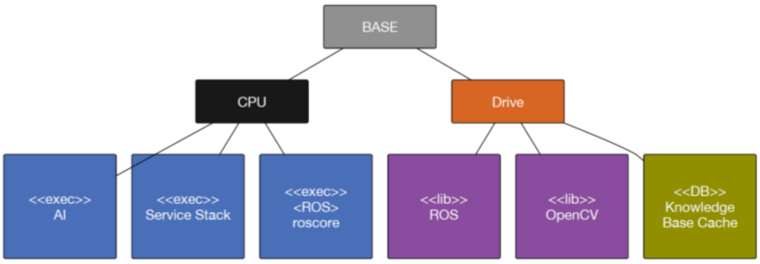
Deployment view

Most specifics of the ERICA's subsystems are unknown at time of writing, thus only a general concept of the deployment can be presented. For the user, two systems are deployed: the mobile system and the base system. Both run UNIX distributions based as ROS relies heavily on an open-source toolchain. The mobile system connects to satellite systems (or subsystems) through a high-speed interconnect, though it is also capable of running packages on its main system. The base system can *only* run package processes, as deployed software is unlikely to require anything other than GPU/CPU computation.



A package here may spawn a process, invoke hardware, and use a library or a database so therefore the deployment view is highly dependent on the actual implementation of these packages. Examples of deployed packages and spawned processes are given below for the mobile system (top) and base system (down).





Hardware requirements

ERICA is designed to be used with common, off-the-shelf hardware (COTS). The mobile system will at its core use a multi-processor, multi-core x86 system connected via Ethernet or other high-speed interconnect to other (vendor specific) nodes. The base system must be horizontally scalable through the connection of heterogeneous compute nodes, for instance through a blade system with a mixture of CPU, GPU and IO nodes.