



ROBOT OPERATING SYSTEM

Παρουσίαση για το μάθημα της Τεχνολογίας Λογισμικού
Δημητριάδης Αλέξανδρος

Τι είναι το ROS;

In a nutshell

Τι είναι το ROS;

ROBOT
OPERATING
SYSTEM

- Το ROS στην πραγματικότητα **δεν** είναι ένα operating system όπως αυτά που γνωρίζουμε αλλά ένα open source meta-operating system για robot.

Τι σημαίνει αυτό;

- Εκτελείται πάνω σε άλλο operating system (Linux).
- Παρέχει τις λειτουργίες που θα παρείχε και ένα κλασικό λειτουργικό σύστημα, όπως για παράδειγμα:
 - hardware abstraction
 - low-level device control
 - message-passing between processes
 - κ.α.
- Ταυτόχρονα όμως διαφέρει από αυτά. (microkernel, nodes αντί για programs κ.α)
- Παρέχει επίσης εργαλεία και βιβλιοθήκες για την ανάπτυξη και την εκτέλεση κώδικα σε διαφορετικού υπολογιστές.

Τι είναι το ROS;

ROBOT OPERATING SYSTEM

- Το ROS στην πραγματικότητα **δεν** είναι ένα operating system όπως αυτά που γνωρίζουμε αλλά ένα meta-operating system για robot.
- Επομένως θα μπορούσαμε πούμε ότι το ROS είναι ένα **middleware** σχεδιασμένο για τη ανάπτυξη ρομποτικών εφαρμογών.
- Είναι **open-source**.
- Δημιουργήθηκε το 2006 στο Stanford University από το εργαστήριο τεχνητής νοημοσύνης του πανεπιστημίου.
- Από το 2008 η ανάπτυξη συνεχίζεται από το ίδρυμα ερευνών Willow Garage με τη συνεργασία πολλών άλλων ιδρυμάτων.

Για πιο λόγο δημιουργήθηκε;

Τι γινόταν πριν από αυτό;

- Για κάθε project ρομποτικής αναπτυσσόταν από την software control system το οποίο ήταν απολύτως εξειδικευμένο στο ζητούμενο task κάθε φορά και αποτελούσε ιδιοκτησία του ατόμου/εταιρίας που το ανέπτυσσε.
- Δημιουργούνταν διαφορετικά συστήματα ακόμα και για παρόμοιους σκοπούς.
- Δεν υπήρχαν standards στη ανάπτυξη τέτοιου είδους λογισμικού.
- Η διαδικασία ανάπτυξης των συστημάτων ήταν πολύ χρονοβόρα και δύσκολη αφού οι developers έπρεπε να ανακαλύπτουν τον τροχό για κάθε project.
- Αποτέλεσμα και η εξέλιξη στον τομέα της ρομποτικής να είναι αργή.

Ποιές αλλαγές έφερε το ROS;

- Δημιούργησε μέσα από συλλογική προσπάθεια έναν αξιόπιστο, **γενικού σκοπού** μηχανισμό για τη δημιουργία ρομποτικών εφαρμογών.

Aerial



Component



Ground



Manipulator



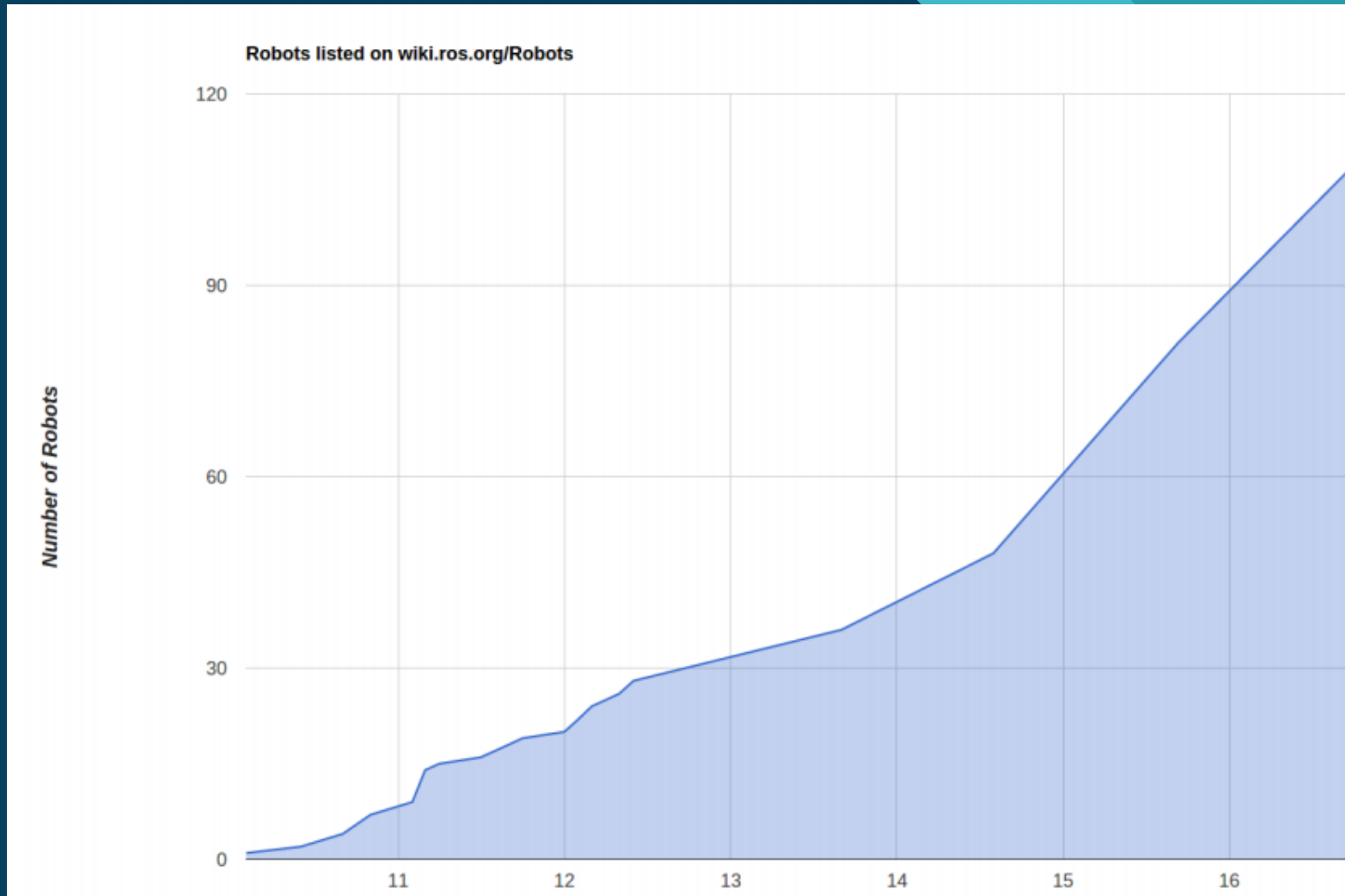
Marine



Διαφορετικές κατηγορίες ρομπότ που υποστηρίζονται από το ROS

Ποιές αλλαγές έφερε το ROS;

- Δημιούργησε μέσα από συλλογική προσπάθεια έναν αξιόπιστο, **γενικού σκοπού** μηχανισμό για τη δημιουργία ρομποτικών εφαρμογών.
- Συγκέντρωσε μία συλλογή από βιβλιοθήκες και εργαλεία τα οποία απλοποίησαν και διευκόλυναν τη διαδικασία παραγωγής σύνθετου λογισμικού για μία ευρεία γκάμα από διαφορετικές ρομποτικές πλατφόρμες που συνεχίζει να μεγαλώνει.



Ο αριθμός των διαφορετικών τύπων ρομπότ που υποστηρίζονται από το ROS τα τελευταία χρόνια

Ποιές αλλαγές έφερε το ROS;

- Δημιούργησε μέσα από συλλογική προσπάθεια έναν αξιόπιστο, **γενικού σκοπού** μηχανισμό για τη δημιουργία ρομποτικών εφαρμογών.
- Συγκέντρωσε μία συλλογή από βιβλιοθήκες και εργαλεία τα οποία απλοποίησαν και διευκόλυναν τη διαδικασία παραγωγής σύνθετου λογισμικού για μία ευρεία γκάμα από διαφορετικές ρομποτικές πλατφόρμες που συνεχίζει να μεγαλώνει.
- Διευκόλυνε και ενθάρρυνε τη συνεισφορά και τη συνεργασία μεταξύ των robot developers με τη δημιουργία ενεργού community και repositories.
- Παρέχει αξιόπιστες και δοκιμασμένες λύσεις σε βασικά προβλήματα του ελέγχου ρομπότ όπως SLAM , Motion Planning , Sensor Integration μέσα από μια μία πολύ μεγάλη ποικιλία από packages τα οποία μπορεί να κατεβάσει και να συμπεριλάβει στο πρόγραμμά του δωρεάν ο καθένας. Αυτό έκανε το robot development πιο εύκολο γρήγορο και αξιόπιστο.
- Δημιούργησε κάποια προγραμματιστικά standards και καλές πρακτικές στο robot development.

Βασικές Αρχές

Βασικές Αρχές

Peer to peer

- Ένα σύνθετο ρομποτικό σύστημα μπορεί να αποτελείται από διάφορους onboard υπολογιστές, development boards, και microcontrollers. Μερικές φορές για πιο απαιτητικά computation tasks χρησιμοποιούνται και offboard computers.
- Γι' αυτό το λόγο υλοποιήθηκε μία peer to peer αρχιτεκτονική η οποία σε συνδυασμό με ένα buffering system και ένα lookup system επιτρέπει σε κάθε component του συστήματος να επικοινωνεί απευθείας με οποιοδήποτε άλλο, σύγχρονα ή ασύγχρονα ανάλογα με τις ανάγκες του συστήματος.
- Το p2p δίκτυο χρησιμοποιεί το πρωτόκολλο XML-RPC.

Multi-Language

- Το ROS είναι language independent , δηλαδή μπορεί να γίνει implement σε όλες τις μοντέρνες γλώσσες προγραμματισμού.
- Μέχρι στιγμής υπάρχουν 3 βασικές client libraries οι οποίες υλοποιούν το ROS σε python (rospy), LISP(roslib), και C++ (roscpp), η οποία είναι και η πιο ευρέως διαδεδομένη βιβλιοθήκη.
- Παράλληλα υπάρχουν πολλές experimental client libraries υποστηρίζουν γλώσσες όπως java(rosjava), go(rosgo), lua(roslua), haskell(roshask) κ.α.

Βασικές Αρχές

Tools-based

- Το ROS είναι ένα microkernel-based σύστημα το οποίο χρησιμοποιεί μεγάλο αριθμό από μικρά εργαλεία για την υλοποίηση και την εκτέλεση των διαφόρων components.
- Για το χειρισμό των nodes και των messages χρησιμοποιούνται εντολές κάθε μία από τις οποίες είναι ένα executable.
- Βασικό πλεονέκτημα αυτού του σχεδιασμού είναι ότι αν υπάρξει πρόβλημα με ένα executable δεν επηρεάζει τα υπόλοιπα, πράγμα που καθιστά το σύστημα πιο ανθεκτικό και ευέλικτο.

Thin

- Οι drivers και οι αλγόριθμοι που υλοποιούνται από το ROS περιέχονται σε standalone executables.
- Αυτό αυξάνει τη δυνατότητα επαναχρησιμοποίησης τους και κρατάει το μέγεθός τους μικρό, ενώ διευκολύνει το unit testing.
- Επίσης δίνει τη δυνατότητα εύκολου integration με άλλα robot software frameworks όπως openRAVE, OpenCV, Gazebo κ.α.

Concepts που υλοποιεί

3 Levels of Concepts

1. ROS Filesystem Level

Περιλαμβάνει τα resources του ROS και την ιεραρχική δομή τους.

2. ROS Computation Graph Level

Περιέχει τα συστατικά του peer-to-peer δικτύου από διεργασίες οι οποίες επεξεργάζονται ταυτόχρονα δεδομένα.

3. ROS Community Level

Περιλαμβάνει τα resources του ROS που επιτρέπουν σε διάφορα communities να ανταλλάσσουν λογισμικό και να επικοινωνούν.

ROS Filesystem Level (I)

- **packages:** . Η βασική μονάδα οργάνωσης του λογισμικού στο ROS. Ένα package ουσιαστικά είναι ένα directory το οποίο περιέχει nodes (ROS runtime processes), external libraries, data, configuration files. Στόχος κάθε πακέτου είναι να παρέχει χρήσιμες λειτουργίες αλλά με εύχρηστο τρόπο που να διευκολύνει την επαναχρησιμοποίησή του.

"Goldilocks" principle: enough functionality to be useful, but not too much that the package is heavyweight and difficult to use from other software

Python: `import rospkg`

C++: `#include <ros/package.h>`

- **repositories (stacks):** Ένα directory το οποίο περιέχει μία συλλογή packages με το ίδιο VCS τα οποία σαν σύνολο προσφέρουν μια λειτουργικότητα.

“navigation stack”: `base_local_planner`, `costmap_2d`, `map_server`

ROS Filesystem Level (II)

- **Package manifests:** xml αρχεία (package.xml) παρέχουν metadata για ένα package, όπως το όνομά του, την έκδοση του, περιγραφή, πληροφορίες για την άδεια, εξαρτήσεις, και άλλες πληροφορίες

```
package format="2">
  <name>foo_core</name>
  <version>1.2.4</version>
  <description>
    This package provides foo capability.
  </description>
  <maintainer email="ivana@osrf.org">Ivana Bildbotz</maintainer>
  <license>BSD</license>
</package>
```

ROS Filesystem Level (III)

- **message (msg) types:** αρχείο με format `.msg` το οποίο ορίζει τη δομή των μηνυμάτων που ανταλλάσσουν τα nodes και περιέχεται στο subdirectory `msg` του κάθε πακέτου. Το ROS περιλαμβάνει εργαλεία που παράγουν αυτόματα κώδικα από το message type σε διάφορες γλώσσες

Πολύ απλή δομή, δυο μέρη: `fields` και `constants`

- **service (srv) types :** αρχείο με format `.srv` το οποίο ορίζει τη δομή των request και response για τα services στο ROS και περιέχεται στο subdirectory `srv` του κάθε πακέτου

ROS Computation Graph Level (I)

- **nodes:** ένα node στο ROS είναι ένα instance ενός executable το οποίο εκτελεί μία διεργασία. Πολλά nodes μπορούν να εκτελούνται παράλληλα και να ανταλλάσσουν μεταξύ τους δεδομένα σύγχρονα ή ασύγχρονα. Κάθε node όταν ξεκινάει να εκτελείται δηλώνει τον εαυτό του στο master. Γράφονται με τη χρήση των client libraries που είδαμε νωρίτερα.

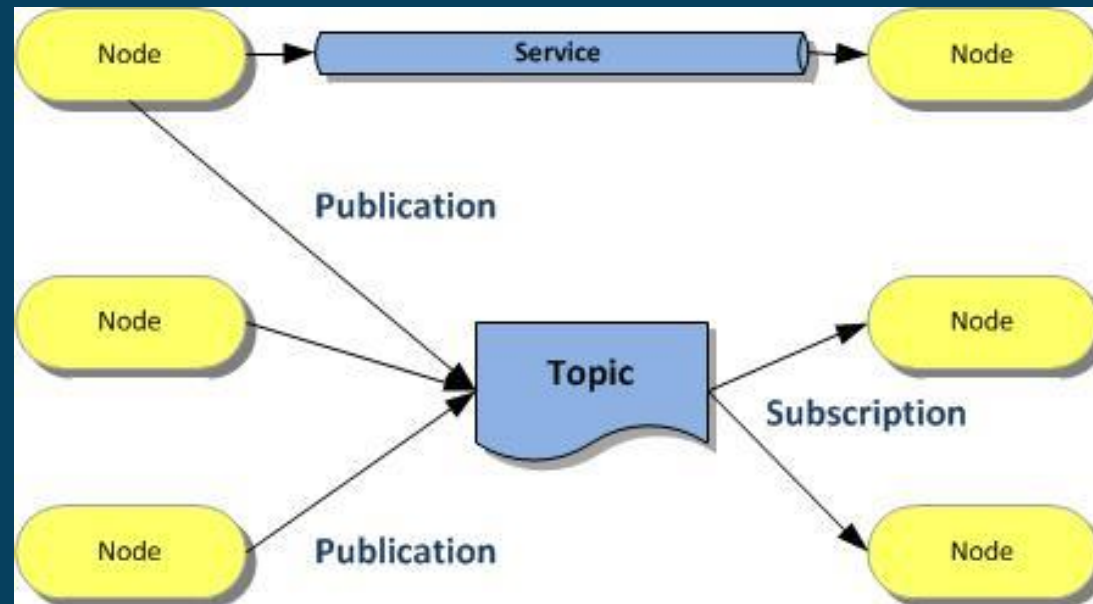
Για παράδειγμα ένα node μπορεί να επεξεργάζεται τα δεδομένα από αισθητήρες laser, ένα να ελέγχει την ταχύτητα των τροχών , ένα να εκτελεί path planning και όλα μαζί να επικοινωνούν για την κίνηση ενός ρομπότ.

- **master:** λειτουργεί ως node declaration και registration service, δίνοντας τη δυνατότητα σε κάθε node να εντοπίζει τα υπόλοιπα ώστε να μπορεί να υλοποιηθεί η p2p επικοινωνία. Αποθηκεύει πληροφορίες σχετικά με τα topics και τα services στα οποία συμμετέχει κάθε node. Χρησιμοποιεί XMLRPC.

Δε εμπλέκεται στην επικοινωνία των Nodes απλά παρέχει lookup information

ROS Computation Graph Level (II)

- **Parameter server:** αποτελεί ένα dictionary στο οποίο αποθηκεύονται πληροφορίες στις οποίες έχουν πρόσβαση τα nodes κατά το runtime. Είναι μέρος του master.
- **messages:** χρησιμοποιούνται για επικοινωνία μεταξύ των nodes, περιέχει primitive τύπους και άλλα messages.
- **topics:** είναι ένα μέσο **ασύγχρονης** επικοινωνίας μεταξύ των nodes, το οποίο βασίζεται σε ένα subscribe/publish σύστημα. Ένα node μπορεί να δημοσιεύσει (publish) δεδομένα στέλνοντας ένα μήνυμα σε ένα topic. Τα υπόλοιπα nodes που έχουν εγγραφεί σε αυτό το topic θα λάβουν το συγκεκριμένο μήνυμα. Όλα τα δεδομένα που στέλνονται σε ένα topic πρέπει να έχουν την ίδια δομή. Λειτουργεί σαν ένα strongly typed message bus.
- **services:** είναι ένα μέσο **σύγχρονης** επικοινωνίας μεταξύ δύο nodes. Χρησιμοποιείται για request/reply interactions.



Επικοινωνία μεταξύ των Nodes στο ROS

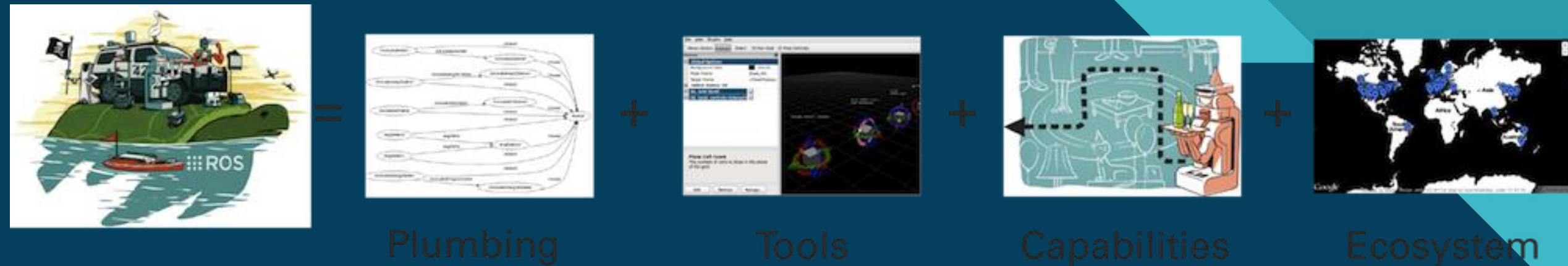
ROS Computation Graph Level (III)

- **bags:** ένας μηχανισμός αποθήκευσης και αναπαραγωγής δεδομένων από ROS messages. Συλλέγει δεδομένα όπως μετρήσεις από αισθητήρες και τα αναπαράγει προσομοιώνοντας αληθινά δεδομένα, λειτουργία ιδιαίτερα χρήσιμη για debugging και testing ενός συστήματος.

ROS Community Level

- **Distributions:** συλλογές από versioned stacks που μπορούν να εγκατασταθούν. Παρόμοια με τα Linux distributions.
- **Repositories:** Το ROS βασίζεται σε ένα ομοσπονδιακό δίκτυο από code repositories στο οποίο μπορούν πολλοί οργανισμοί να αναπτύξουν και να εκδώσουν τα δικά τους robot software components.
- **The ROS Wiki:** Το βασικό φόρουμ για documentation και πληροφορίες, tutorial κ.α σχετικά με το ROS. Ο καθένας μπορεί να συνεισφέρει.
- **Bug Ticket System:** πλατφόρμα αναφοράς bugs και αιτημάτων για νέα features σχετικά με το ROS.
- **Mailing Lists:** κανάλι επικοινωνίας και αποριών μεταξύ των χρηστών.

Συνοψίζοντας μπορούμε να περιγράψουμε το ROS με την παρακάτω εξίσωση:



$ROS = plumbing + tools + capabilities + ecosystem$

Πηγές

- <https://answers.ros.org>
- https://en.wikipedia.org/wiki/Robot_Operating_System#2007
- <https://robohub.org/ros-101-intro-to-the-robot-operating-system/>
- <http://www.ros.org/about-ros/>
- <https://www.generationrobots.com/blog/en/ros-robot-operating-system-2/>
- <https://www.toptal.com/robotics/introduction-to-robot-operating-system>
- <http://wiki.ros.org>

The background features a diagonal design. The upper-left portion is a light blue triangle. The lower-right portion is a dark blue area containing the text. A thin, dark blue diagonal line separates the light blue area from the dark blue area. The text is white and centered in the dark blue area.

Ευχαριστώ για
το χρόνο σας