H2020 funded
GA no. 732287

rosin-project.eu

ROS-I Academy Training

ROS Introduction

MASCOR Institute
FH Aachen University

2017ff

FH AACHEN
UNIVERSITY OF APPLIED SCIENCES

ROSIN Training Robot Operating System

- ROS celebrates its 10th birthday this year!

https://vimeo.com/245826128

ROS Introduction

ROS – Robot Operating System

- Operating System?
- Accumulation of programs
- Installation of third party programs
- File System
- Hardware drivers
- Programming environment

- Robot Framework?
- Preinstalled OS necessary
- Similar to other Robot Frameworks
  - Microsoft Robotics Studio
  - Player
  - Fawkes

ROS as a Meta-Operating System / Middleware

ROS Benefits

ROS – Robot Operating System

- Standard in mobile robotics
- Open Source
- Big Community
- Reuseability of software
- Fine grained software packages
- Network based communication
- Decentralized system

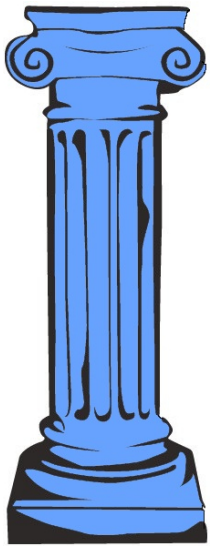
ROS Benefits

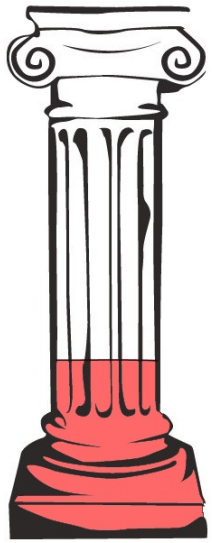
ROS – Robot Operating System

- Hardware driver
- Complete supported robotsystems (125)
- Programming Languages: C++, Python, Lua, Java. . .
- Integration of nearly all other Open Source libraries
  - OpenCV - image processing
  - Gazebo - simulation environment
  - MoveIt! - pathp lanning
  - PCL - 3D pointcloud library
  - **ROS Industrial**

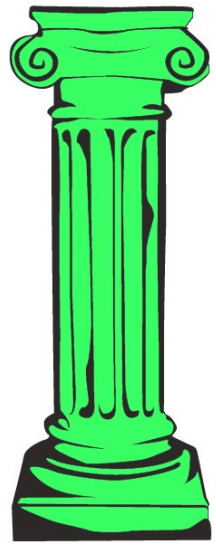
ROS
Level of Concepts

Higher-Level Concepts





www.wadeco.de



Filesystem

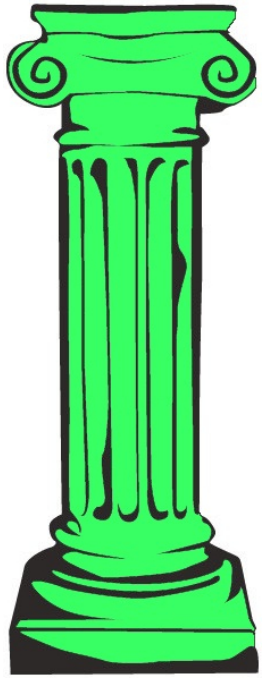
Computation Graph

Community

ROS Community

The first level...

... Community



www.wadeco.de

ROS Community Worldwide

H2020 funded
GA no. 732287

http://www.ros.org/is-ros-for-me/

- ROS has grown to include a large community of users worldwide
- The Community is the major pillar of ROS

ROS Community Resources

ros.org

ROS Website

ROS Answers
&
ROS Discourse

ROS Wiki

ROS News

ROS Community
Some Facts: Users (July 2015)

H2020 funded
GA no. 732287

wiki.ros.org

1.	United States	69,989 (24.17%)
2.	China	31,778 (10.97%)
3.	Germany	27,080 (9.35%)
4.	Japan	22,867 (7.90%)
5.	India	9,740 (3.36%)
6.	France	9,431 (3.26%)
7.	United Kingdom	9,143 (3.16%)
8.	Spain	8,617 (2.98%)
9.	Canada	8,497 (2.93%)
10.	Italy	7,176 (2.48%)

- 16,043 wiki pages
- 37,235 views / day
- 46,611 unique IP addresses downloading .deb files

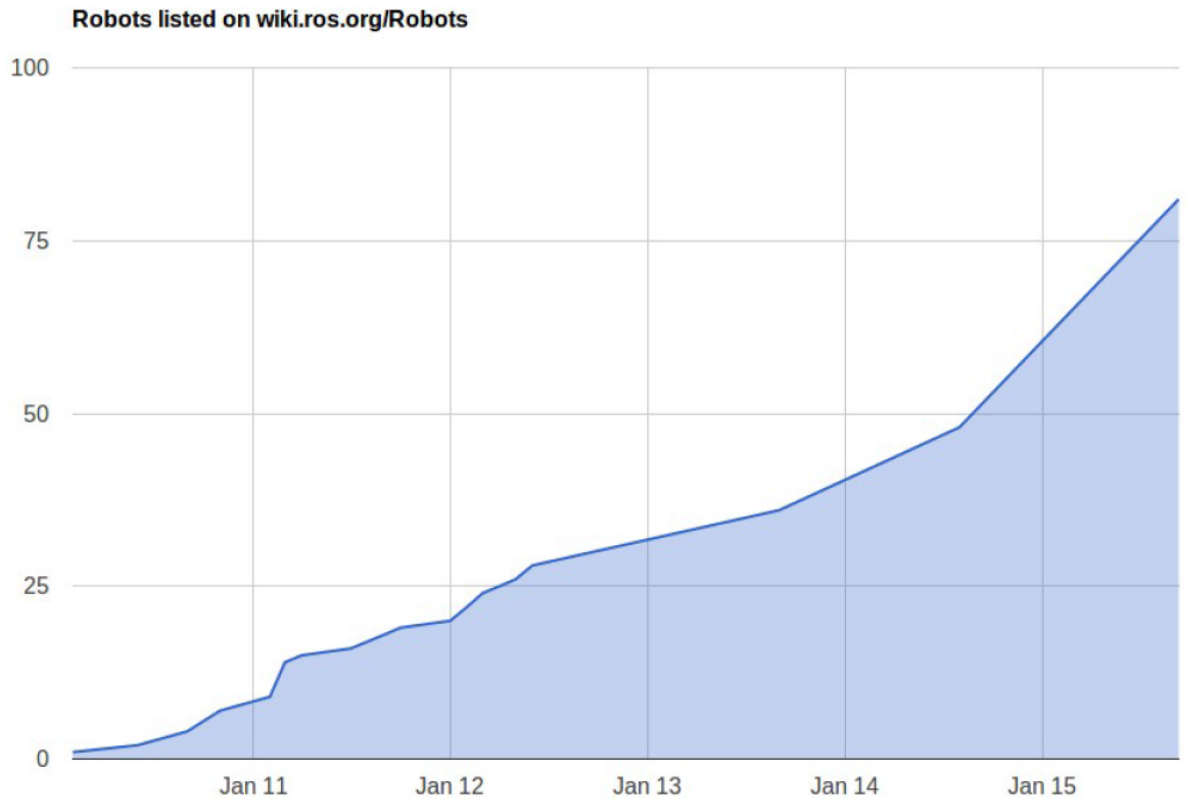
answers.ros.org

	Page	Pageviews	% Pageviews
1.	/questions/	14,708	4.98%
2.	/question/87866/how-to-edit-the-bashrc-file/	2,625	0.89%
3.	/questions/ask/	1,173	0.40%
4.	/account/signin/?next=/	1,155	0.39%
5.	/question/203610/ubuntu-14042-unmet-dependencies/	1,025	0.35%
6.	/account/signin/	931	0.32%
7.	/question/10543/how-do-i-link-a-library-in-cmake/	773	0.26%
8.	/question/196455/kinect-installation-and-setup-on-ros-updated/	645	0.22%
9.	/questions/scope:all/sort:activity-desc/page:1/	582	0.20%
10.	/question/57213/how-i-completely-remove-all-ros-from-my-system/	518	0.18%

- 24,026 total questions
- 17,414 answered questions
- 523 new questions

Community Metrics Report, Tully Foote, July 2015

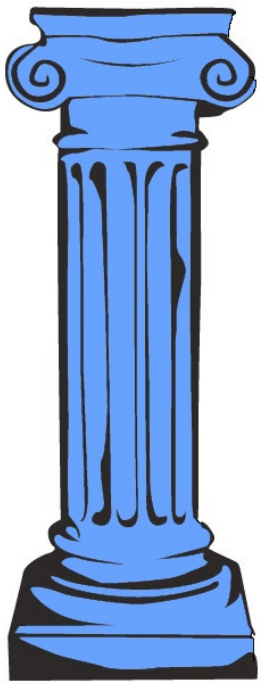
ROS Community
Some Facts: Research (July 2015)



- Number of papers citing “ROS: an open-source Robot Operating System” (Quigley et al., 2009):
  1843 (**52%** increase)

Community Metrics Report, Tully Foote, July 2015

ROS Filesystem



www.wadeco.de

The next level...

... Filesystem

ROS Filesystem Package

- Software in ROS is organized into **packages**
- Smallest build part in ROS
- Dedicated to one functionality, e.g.:
  - Hardware driver
  - Algorithm
  - Visualization tool
  - Library . . .
- Packages can be grouped to **metapackages**
  - References one or more related packages
  - Beneficial for release and versioning
  - (Former called stacks)

`REP-144:ROSPackageNaming`

ROS Filesystem
Package install options

Debian Packages:

- Automatic installation
- Stable versions
- Prebuilt binaries

Source Repositories:

- “Latest” code
- Manual compilation
- Allows code adjustments

ROS Filesystem
Package installation (Debian)

- Debian:

sudo apt-get install ros-kinetic-package_name

admin permission	advanced packaging tool	install new package	ROS package	ROS distribution	ROS package name

- Automatic installation:
  - Location: /opt/ros/<distro>/. . .
  - Installs all required dependencies

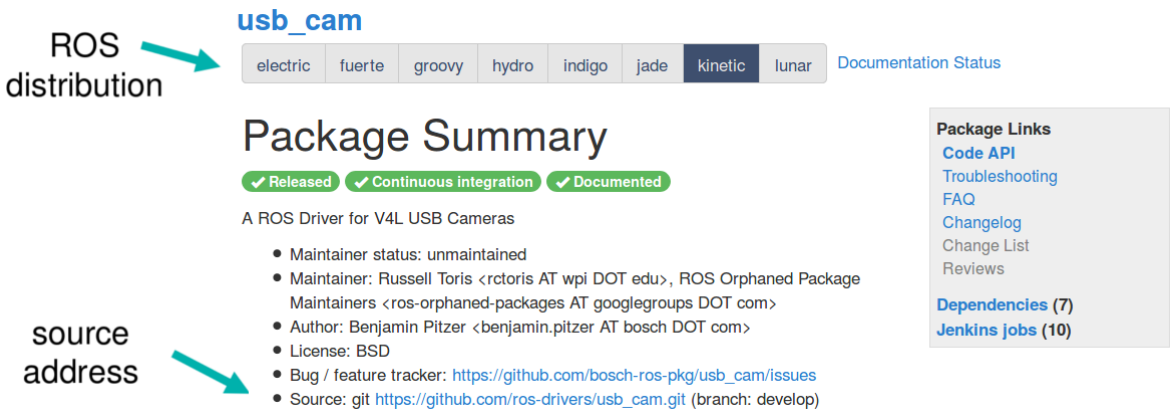
- To remove a package:

sudo apt-get remove ros-kinetic-package_name

ROS Filesystem
Package installation (Source)

- Download from source (usually git)

git clone source-address



- Manual compilation via *catkin* (Explained later)
- To remove a package:
  Delete package folder and re-compile
- Manage multiple version-control packages: `wstool`

ROS Filesystem
Package content - Folder structure

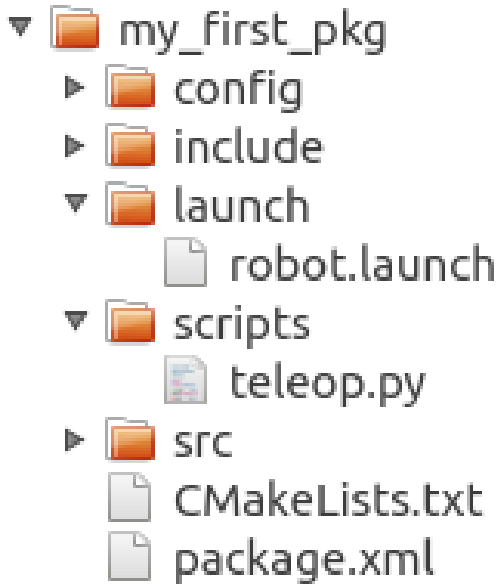
/src – Compileable files (**C++**)

/include – Header files for C++

/scripts – Scripts (**Python** or Shell)

/launch – ROS ***launch*-files** (Explained later)

/config – Configuration file (e.g. yaml)



- Command to generate a new package:

catkin_create_package package_name dependencies

ROS Filesystem
Package content - Manifests (package.xml)

- Meta information about the package
- Lists dependencies of package
- *Format 2* is recommended (Old *Format 1* still supported)

Format

Meta information

```
<package format="2">
  <name>your_package</name>
  <version>1.2.4</version>
  <description>
    This package adds extra features to rosawesome.
  </description>
  <maintainer email="you@example.com">Your Name</maintainer>
  <license>BSD</license>
  <buildtool_depend>catkin</buildtool_depend>
</package>
```

Dependencies

ROS Filesystem Dependencies

- Dependencies can be
  - Other ROS packages
  - System libraries (e.g. “Boost”)
- Format 2 dependencies (`Catkin-howto`):

<buildtool_depend> – Required for build tools (*catkin* is mandatory)

<exec_depend> – Required for execution (e.g. Python script ...)

<build_depend> – Required for building (e.g. C++ library ...)

<test_depend> – Required for testing

<build_export_depend> – If exported header depends on other ones

<depend> – All-in-one (If **all** are required)

ROS Filesystem
Package content - CMakeLists.txt

- Defines build rules for catkin. E.g.:
  - Declare compilation of executables
  - How to resolve header and library references
- Mostly Cmake, plus catkin-specific ones
- Does not know about package.xml dependencies

```
cmake_minimum_required(VERSION 2.8.3)
project(your_package)
```

Package name

```
find_package(catkin REQUIRED COMPONENTS ...)
...
catkin_package(...)
```

CMake rules

ROS Filesystem
Catkin

- ROS build system:
  - Based on CMake macros and Python scripts
  - Cross-platform (Ubuntu, Windows, Embedded-Linux)
- ROS packages are managed via “workspaces”
  - Catkin can create and compile them

Catkin (Used in training):

- Command:

  catkin_make

- Must be called in the root of the workspace

Catkin tools:

- Command:

  catkin build

- Builds each package in isolation

ROS Filesystem
Catkin workspace (1)

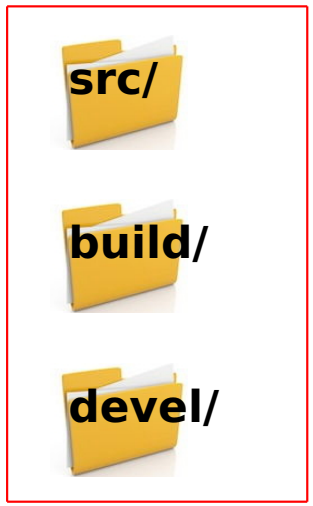
- Folder where you modify, build and install catkin packages

ROS Filesystem
Catkin workspace (2)

Used for Development

- Source Space:
  - Contains the source version of packages
- Build Space:
  - Where CMake is invoked and generates artifacs
- Devel Space:
  - Where built targets are placed prior to installation

Dependend



Ready for Installation

- Install Space:
  - Self-contained package ready for release
  - Package-structure like in */opt/ros/<distro>*

Self-contained

ROS Filesystem
Compile workspace

Example with catkin_make

- Go to the ROS workspace

```
cd ~/catkin_ws
```

- Install all dependencies of package(s)

```
rosdep install -i --from-paths src
```

- Compile the workspace

```
catkin_make
```

ROS Filesystem
Environment Setup File

setup.sh	–	Environment setup in general
setup.bash	**–**	**Environment setup file for Bash**
setup.zsh	–	Environment setup file for Z shell

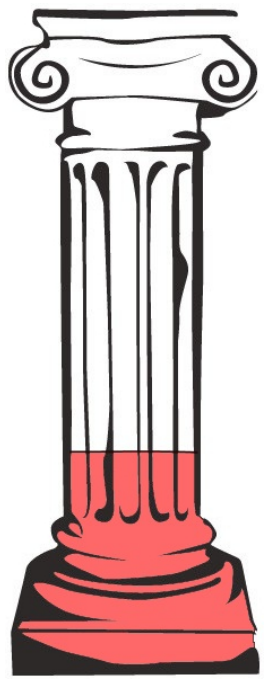
- Generated during the initialization process of a workspace
- Extends the present ROS environment

→ ROS can find and use any resources that have been installed or built inside the ROS environment

- Source ROS env: **source <ws-path>/devel/setup.bash**
- Check current env: **echo $ROS_PACKAGE_PATH**

ROS Computation Graph

The next level ...

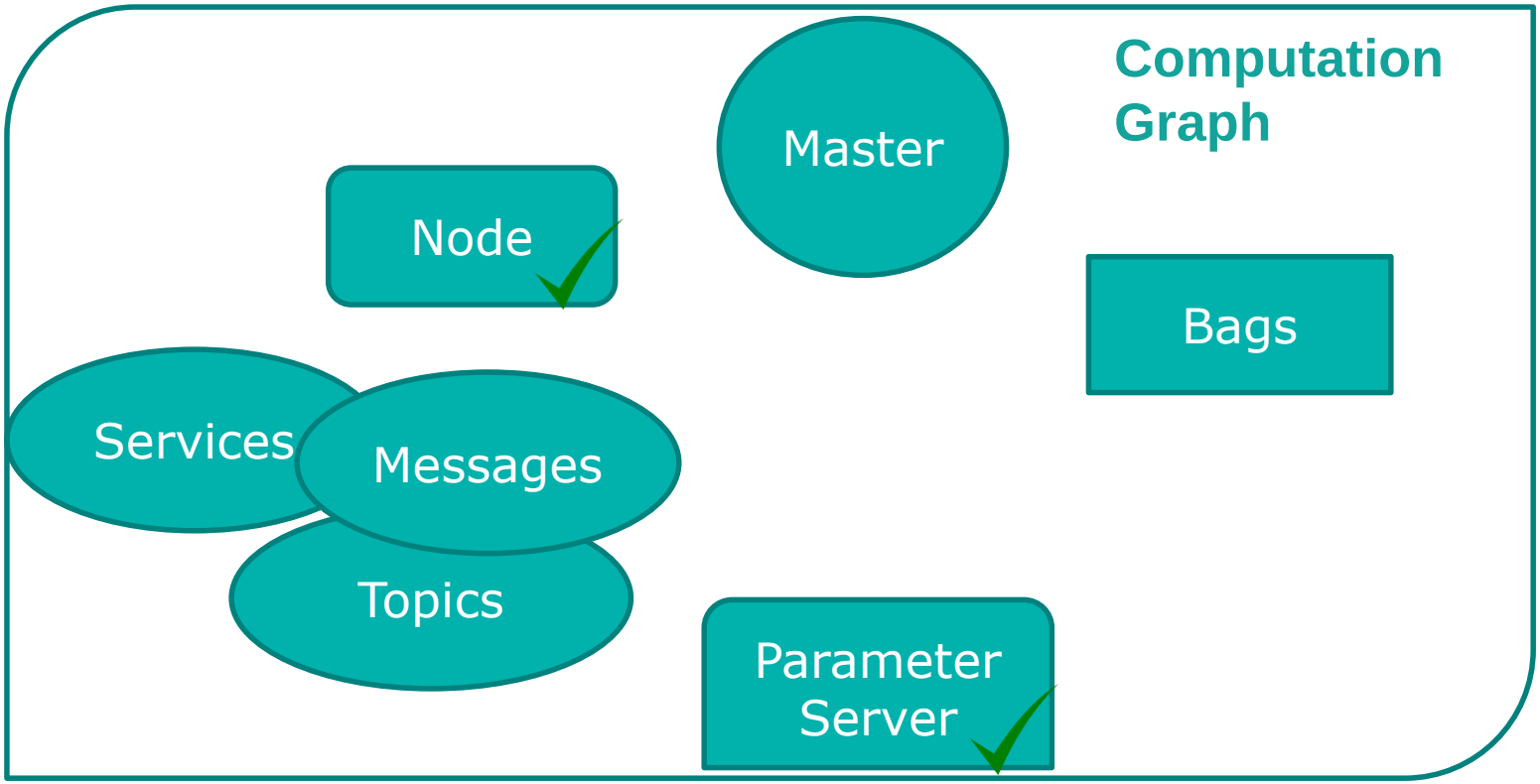


www.wadeco.de

... Computation Graph

ROS Computation Graph Concept

- The *Computation Graph* is the peer-to-peer network of ROS processes that are processing data together.



- Reflects the whole communication in the ROS system.

ROS Computation Graph
Node

- Executable part of ROS:
  - Scripts for Python
  - Compiled source code for C++
  - Generally programs using a ROS client library
    - Haskell, Lisp, Matlab, ...
  - Process that performs computation
  - Meant to operate at fine-grained scale

To run a node:

```
rosrun package_name node_name
```

→ A robot control system will usually comprise many nodes

Path planning Manipulation Motor controller Camera

ROS Computation Graph
Parameter Server

- Allows data to be stored by key in a central location
- Globally viewable
- Not designed for high-performance

⟶ static, non binary data (configuration data)

- Examples
  - adjustable hardware drivers:
    - webcams,
    - joysticks, ...
  - adjustable algorithms:
    - path planning,
    - sensor fusion, ...

⟶ More flexibility

ROS Computation Graph
Launch Files

- XML based
- Starts the **roscore**
- Tool to manage a robotic system:
  - setting of parameter values
  - including other launch files
  - definition of namespaces
  - respawning of died nodes

To run a launch file:

```
roslaunch package_name launch_file_name
```

ROS Computation Graph
Launch Files Example


```
<!-- -*- mode: XML -*- -->
<launch>
<node name="cam" pkg="usb_cam" type="usb_cam_node" output="screen">
    <param name="video_device"   value="/dev/video0" />
    <param name="image_width"    value="640" />
    <param name="image_height"   value="480" />
</node>
<node name="view" pkg="image_view" type="image_view" output="screen" />
</launch>
```

beginn of the launch file

starting a node

setting parameters

ending a start node process

name of running process

name of executable

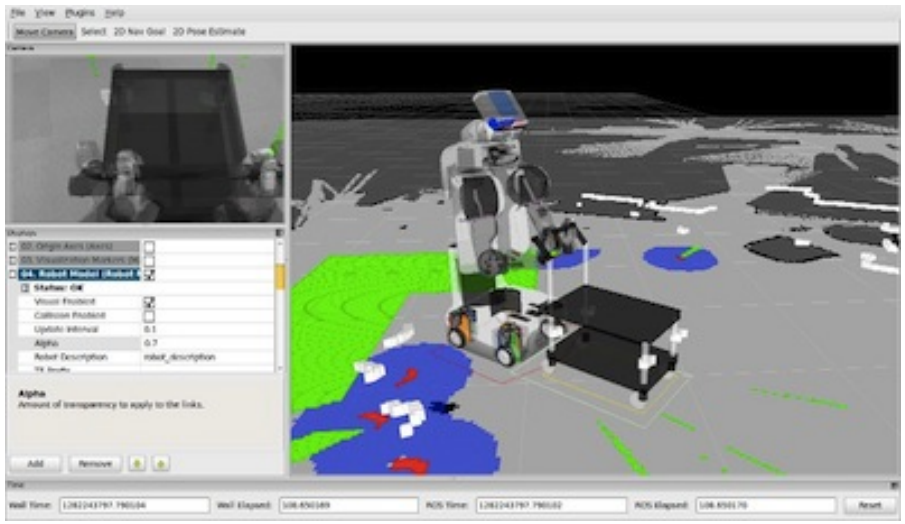
name of package

output of errors and warnings

end of the launch file

ROS Computation Graph
RViz

- 3D visualization tool of ROS
- Can be used to visualize any type of sensor data or algorithm results, e.g.:
  - images,
  - laser scan data,
  - imu data,
  - transformations,
  - maps,
  - robot models,
  - planned paths,
  - etc.



To run RViz: **rosrun rviz rviz**

ROS
Any questions?

http://www.allonrobots.com/