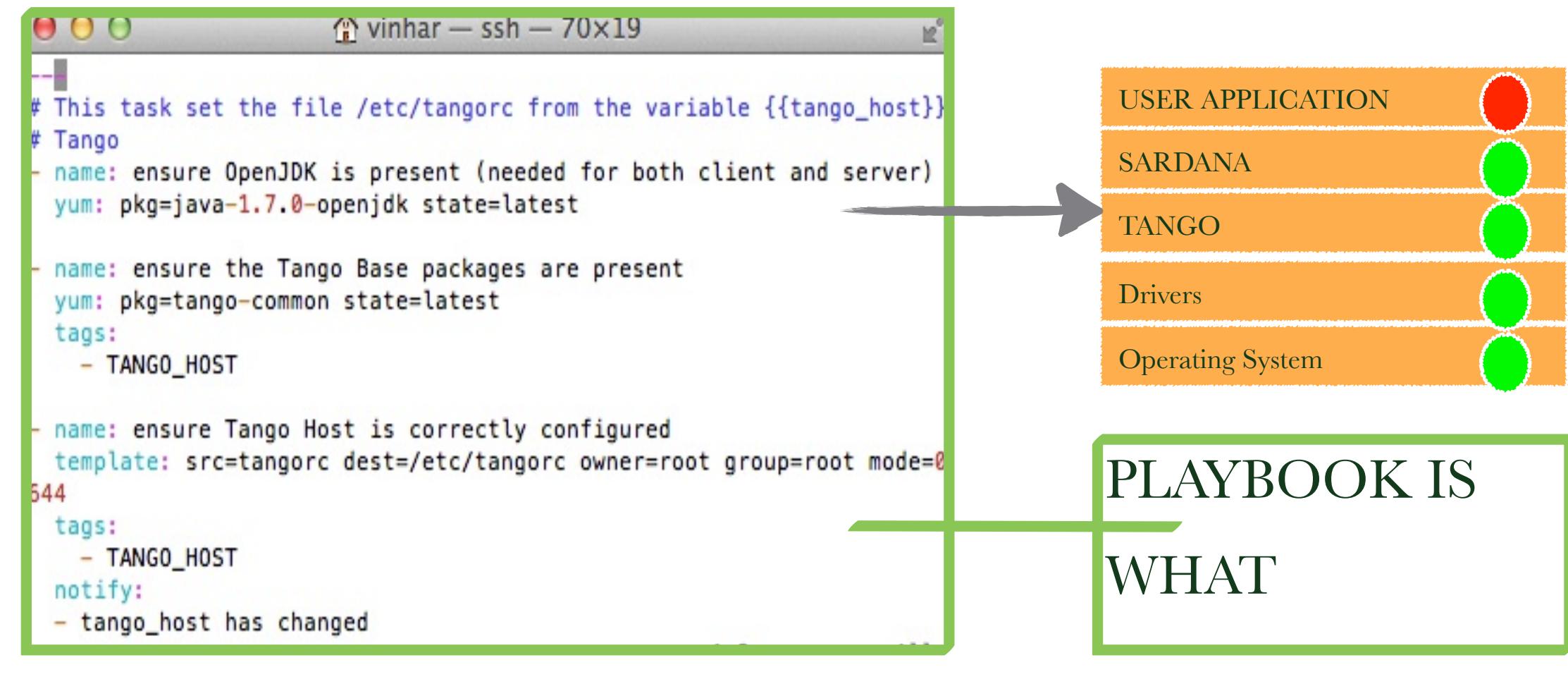


# Configuration Management of the Control System

# Automation as a tool

The first principle for the long term strategy of the Kontroll and IT Support (KITS) group at MAX IV is to develop the expertise and was inspired by the Toyota way[2]. One action involves eliminating repetitive manual interventions which in the end bring little value.



# Configuration of the Control System

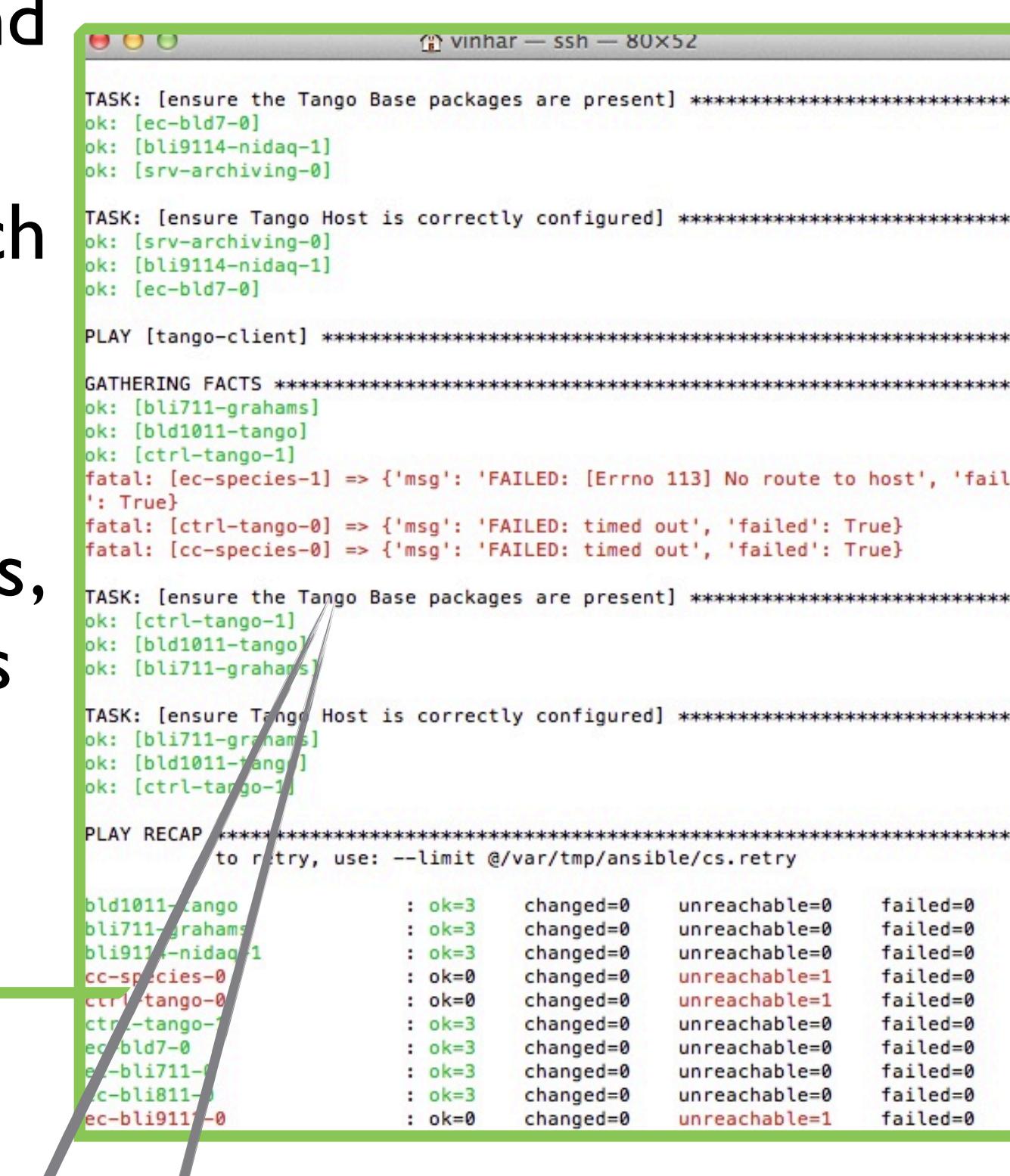
# Operating Systems: CentOS, Ubuntu and Windows

# Network services: LDAP and NFS storage connection

Device drivers and libraries such as Python and Tango

# Tango devices

Applications such as GUIs  
Sardana macros and controllers

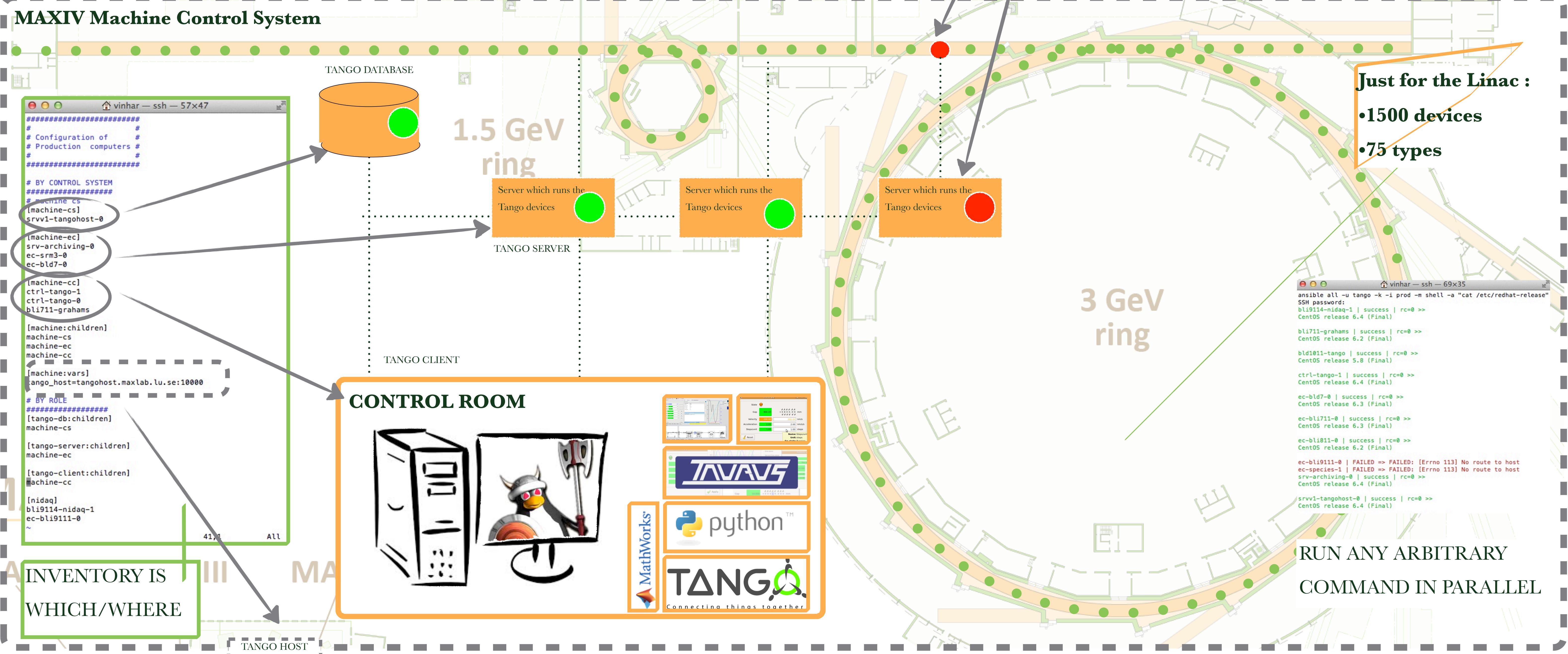


## Configuration in detail

The MAX IV control system is based on Tango for the middleware layer and Sardana for the application layer, with its Taurus Framework as the graphical user interface and additionally spock as the command line interface. Each beamline and the Machine will have their own control system on different separated networks. The linac will be composed by around 1500 Tango devices to control 75 different types of device. A majority of the physical values will be monitored with PLCs and also, most of the controllers will be driven by TCP/IP communication.

The different types of software in the MAX IV Control System consist of a set of operating systems, network services and base frameworks such as Tango, device drivers, libraries, Tango devices and applications such as GUIs, Sardana macros and controllers which all need to be deployed and configured before usage. The configuration is not only necessary for the installation but also during the lifetime of each component when a new version update needs to activate new functionalities.

Any difference of configuration between computers is a potential risk the consequence being to change the behaviour of a piece of software. To avoid these differences, a best practice in the Continuous Integration process is to compile and test in the environment closest possible to the target platform[1]. This can be done by cloning a production server but a preferable method is to track the minimal dependencies and configuration of each piece of software.



# Ansible

**Inventory** keep the computers list and their roles inside the control system like:

- the Tango database,
  - the server which runs the Tango devices,
  - the client computer which runs GUIs and CLIs

**Ad hoc command** is useful to complete an action on several computers in the same time. Ansible uses the inventory file to include the computers in the execution list. Ansible comes with a predefined list of actions to execute a shell command, to install a package, to start a service, etc but it also exists 3rd party modules to manage specific applications like MySQL.

**Playbook** is used to keep the different actions needed to reach a configuration state instead of repeating them manually with an ad hoc command. The playbook is used with the inventory file during the execution of Ansible.

# Properties

**Idempotent**: the capacity to check if a system is compliant with the reference without needing to modify it. The same operation applied several times has the same consequence as if applied once.

**Stateless:** the application should not leave tracks in the operating system to avoid any memory effects. The configuration is held in one place.

**Small Footprint:** the deployment is reduced to a minimum of servers to avoid spending much time managing the configuration management system. The system should come with a minimal dependency set.

# References

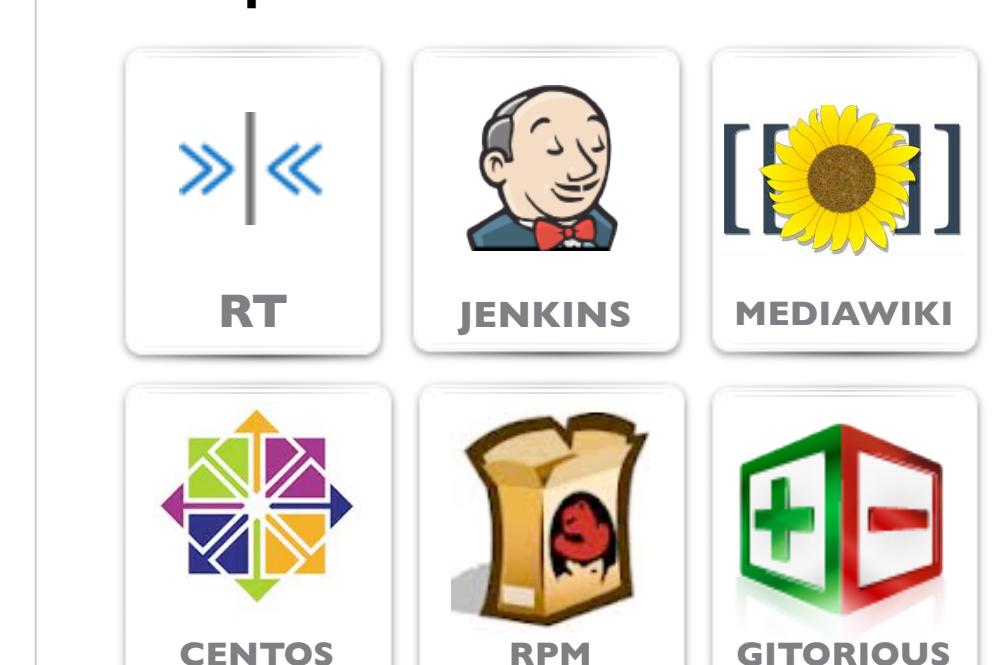
Tango and Sardana: <http://www.tango-controls.org/>

Ansible: <http://www.ansibleworks.com>

## **Advantages in details**

- Greatly reduce the deployment time
  - Keep track of the essential configuration version if a content versioning system is used (GIT, SVN, ...).
  - Keep coherence between configuration data and the process using it by considering the version of a software component as part of the configuration
  - Apply an additional configuration entry while at the same time being able to upgrade to the latest version of the software which is able to process it (e.g. new property for a Tango device server, ...)
  - Make it easy to refactor the data organisation allowing to change by small increments, reducing

## Development Infrastructure



Several Open Source Software compose our toolbox to support the development of the control system.

RT the request tracker gathers the requirements and issues from the users.

Git is used to manage the source code of each project, even the user's program.

- MediaWiki publishes useful information for steering the control system.
- Jenkins automatically builds the last project from the standard platform, ready for deployment
- Finally the project is deployed with all its dependencies via a RPM package