

CIT  EC

Cognitive Interaction Technology  
Exzellenzcluster  
Universität Bielefeld

 CORE Lab

# Modeling Software Systems in Experimental Robotics for Improved Reproducibility

A Case Study with the iCub Humanoid Robot

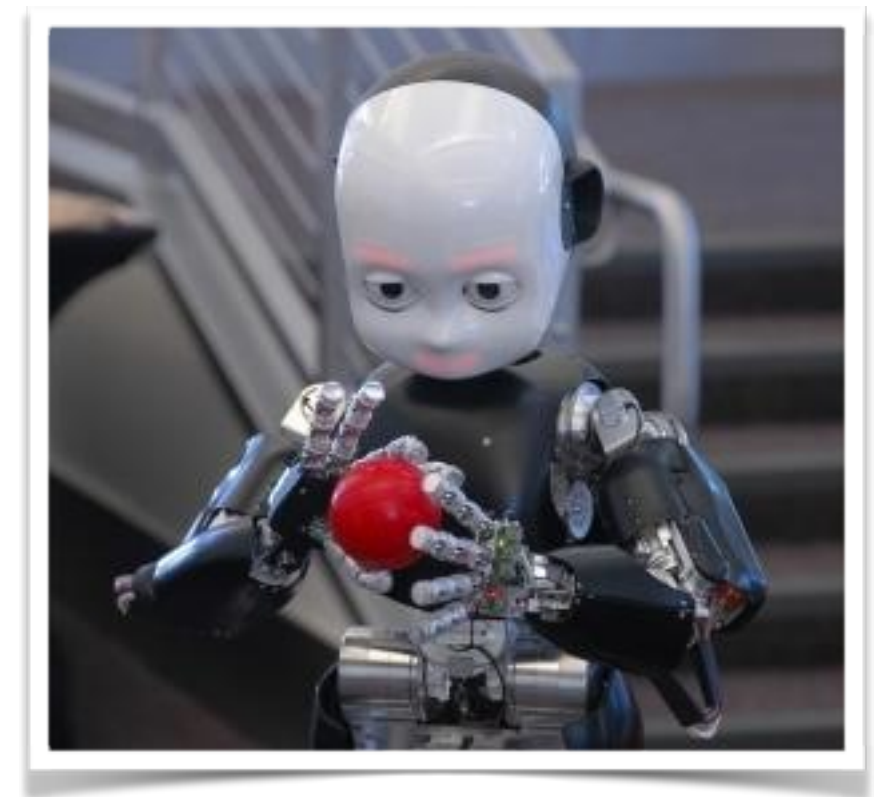
# Systems in Experimental Robotics | **Current State**

Research on autonomous robots achieved considerable progress over the recent years.

Reported results are typically validated through experimental evaluation or demonstrated live at robotics competitions. [ DARPA Challenge, RoboCup, ... ]

Amongst other reasons, this is due to: „off-the-shelf“ robots [ iCub, NAO, ... ] and vivid open source communities providing reusable building blocks, e.g. ROS, Orocos, OPRoS, Yarp, yarp-wholebodyinterface, GURLS, ... ;)

Publicly available data sets are used to improve benchmarking procedures, i.e., Rawseeds Project. [1]



» Numerous studies have been performed over the past 15 years, but one of the hallmarks of science has yet to be achieved: results at present are hardly ever reproducible by other research groups. « [2]

---



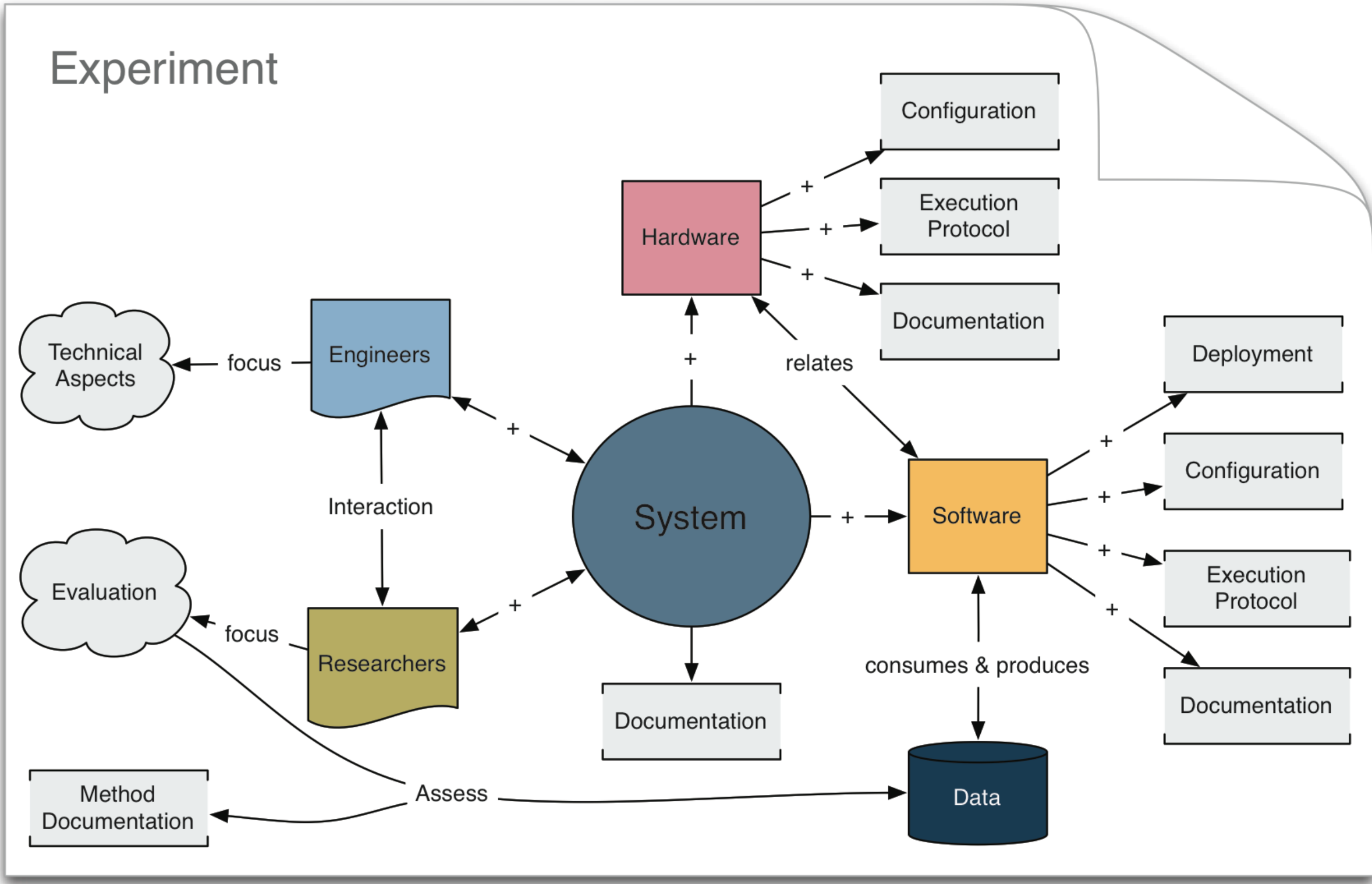
# Systems in Experimental Robotics | **Current State**

Methodological issues that prevent reproducibility of robotic system experiments. [3]

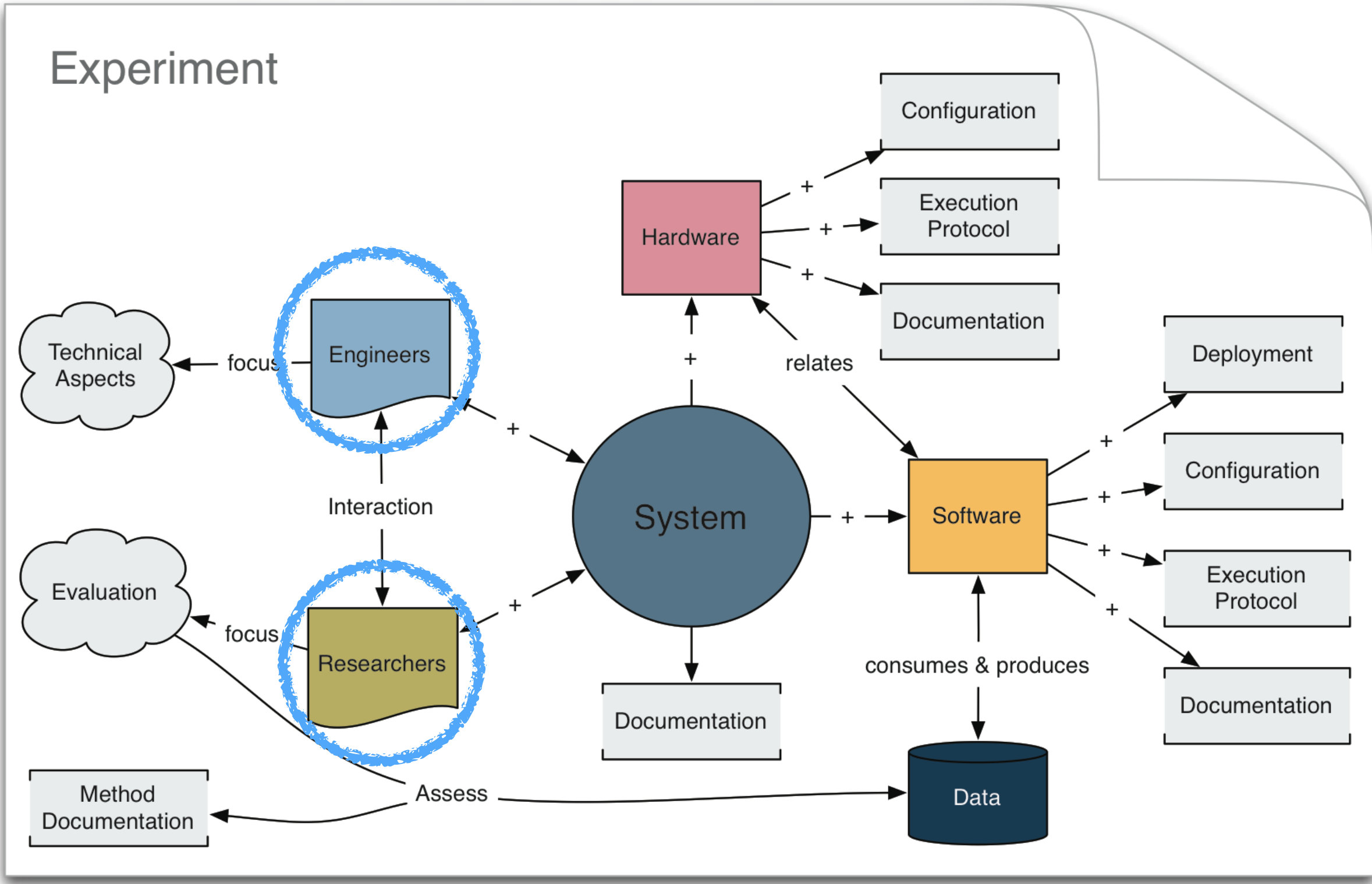
This includes the frequently neglected impact on experiments caused by the relationship between **individual** components and the **whole** system — in component-based systems for instance.

Question: how do publications need to be written and published in order to **improve reproducibility**?

# Systems in Experimental Robotics | Aspect Overview



# Systems in Experimental Robotics | Overview



# Systems in Experimental Robotics | **Problem Statement**

**Information retrieval and aggregation:** artifacts are often distributed over different locations, thus already the discovery, identification and aggregation of all required artifacts is difficult.

**Semantic relationships:** which specific versions (master/v. 133.7) of software components were in use for a particular study?

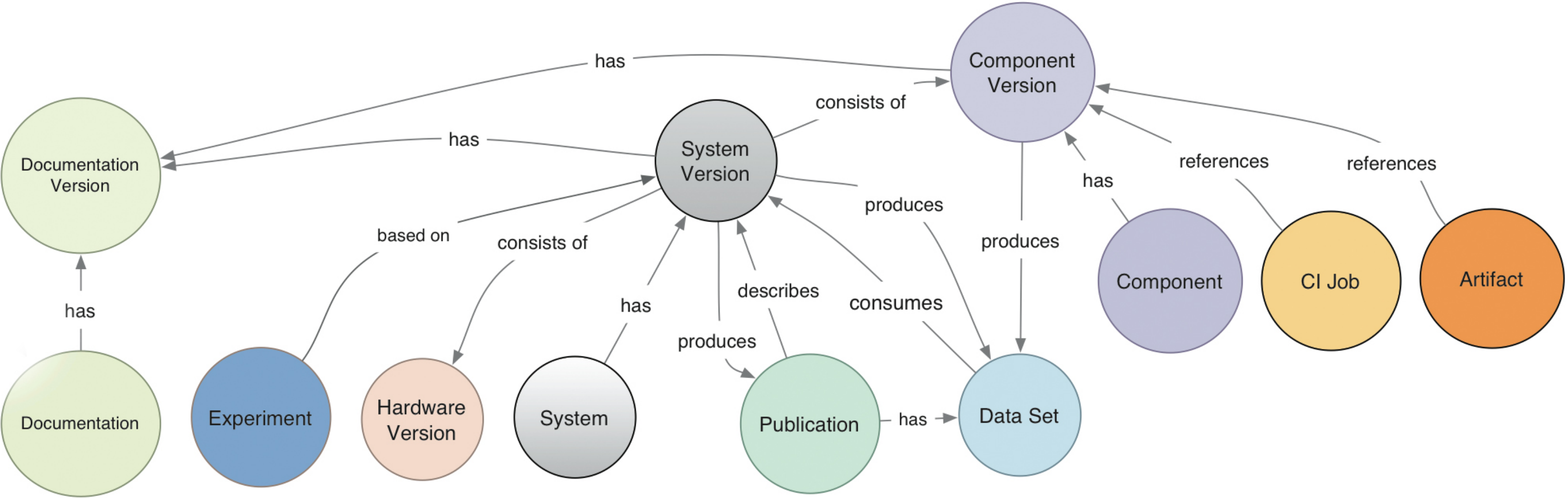
**Software deployment:** Most current systems are realized using a component-based architecture. They do not necessarily use of the same build infrastructure CMake, Catkin, maven, etc., binary deployment mechanism and execution environment.

**Experiment testing, execution and evaluation:** Advanced robotics experiments require significant efforts spent on system development, integration testing, execution, evaluation and preservation of results. This is particular costly if many of these tasks are carried out manually. Crucial run time parameters and component configurations are often omitted or not documented properly.

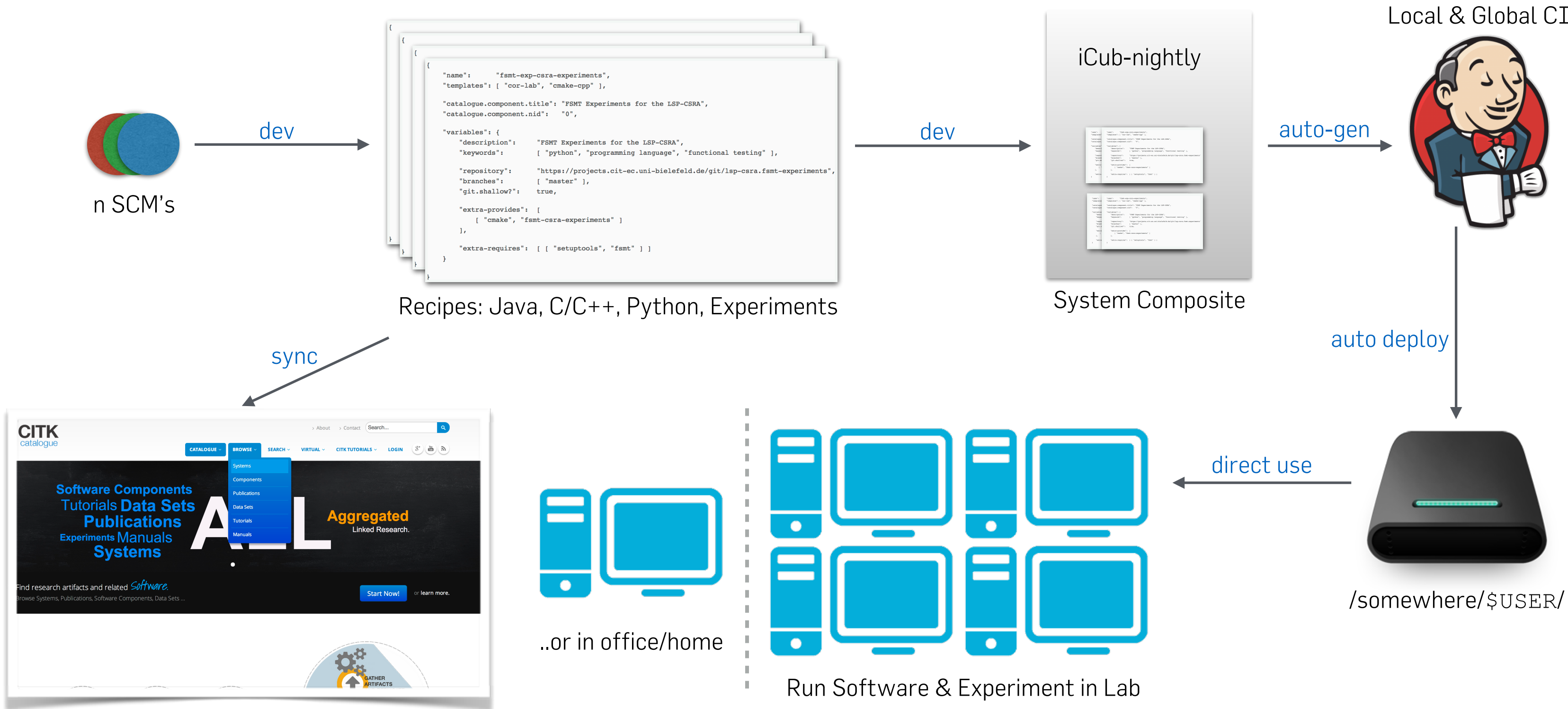
To tackle these issues we introduce an approach for reproducible robotics experimentation based on an **integrated software toolchain** for system **developers** and experiment **designers**.

---

# Systems in Experimental Robotics | System Model



# Systems in Experimental Robotics | Technical Realization Overview



Web-based catalog (Systems and related artifacts)

# Systems in Experimental Robotics | Technical Realization Recipes





# Systems in Experimental Robotics | Technical Realization Recipe (Experiment)



```

1 {
2   "name":      "fsmt-exp-icub-nightly",
3   "templates": [ "cor-lab", "cmake-cpp" ],
4
5   "catalogue.component.title": "FSMT Experiment iCub Ball Tracking",
6   "catalogue.component.nid": "1276",
7
8   "variables": {
9     "description":      "Replay recorded data set and make the icub track a red ball",
10    "keywords":         [ "fsmt", "functional testing", "iCub" ],
11
12    "repository":      "https://openresearch.cit-ec.de/git/fsmt.experiments.git",
13    "branches":        [ "master" ],
14    "sub-directory":   "icub-nightly",
15
16    "extra-requires":  [ [ "setuptools", "fsmt" ] ],
17
18    "redmine.instance": "https://openresearch.cit-ec.de/",
19    "redmine-project": "fsmt"
20  }
21 }
  
```

# Systems in Experimental Robotics | Technical Realization Experiment Protocol

Local & Global C...



external queue internal queue

CITK catalogue

Software Tutorial Pub Experiments Sy

Find research artifacts and re  
Browse Systems, Publications, Softw

..or in office/home



Lab

ity. However, the sponsorship of an automated build infrastructure and tools to automatically create build jobs (cf. ROS-bloom and Section 2.2) reduces the amount of expert knowledge and is thus also considered beneficial. In contrast to our approach, ROS and iCub distributions can be installed via source builds (*not recommended* as stated in the ROS wiki) but also via binary distributions that simplify and speed up installation time. On the other hand, binary packages often raise typical issues such as requiring root permissions for installation, the install prefix is fixed and creating binary packages for diverse operating systems and flavors is a huge effort. With respect to build systems both ecosystems are based on CMake, which facilitates cross-platform compatibility, but also, in contrast to CMake, restricts the number of integrable third-party build tools. This is especially crucial because robotic systems/experiments often incorporate artifacts from more than one ecosystem. Finally, experiment specification, orchestration, automated execution and evaluation is not supported by either ROS or the iCub infrastructure.

✎ In order to *verify* our results, please visit: [iCub Ball Tracking-nightly](#)

## 5 Conclusion

We introduced an approach for reproducible robotics experimentation based on an integrated software toolchain for system developers and experimenters. It combines state-of-the-art technologies into a consistent process that facilitates the reproduction of robotic systems and experiments. We briefly outlined the replication process for a simulation experiment and discussed the benefits of the approach in comparison to well-known robotics ecosystems and their support for reproducible experimentation. Future work will focus on providing the complete toolchain as open source to the community, extending the build generation with

Find CMake link in publication

# Systems in Experimental Robotics | Step by Step

The screenshot shows the CITK catalogue website. The header includes the logo 'CITK catalogue', navigation links for 'About' and 'Contact', a search bar, and a menu with options like 'CATALOGUE', 'BROWSE', 'SEARCH', 'VIRTUAL', 'CITK TUTORIALS', and 'FLIER'S CONTRIBUTIONS'. There are also social media icons for Google+, YouTube, and RSS. The breadcrumb trail is 'Home / Systems / iCub Ball Tracking / Version /'. The main title is 'iCub Ball Tracking-nightly'. Below the title are buttons for 'View', 'Edit', 'Track', and 'Devel'. The 'Version' section lists: 'Version: nightly', 'Acronym: icub-btracking', 'Version of System: iCub Ball Tracking', and 'Videos: iCub Ball Tracking Demo Video'. The 'Description' states that this version uses tools like yarview, iCubGui, and dataSetPlayer to replay pre-recorded sequences. It encourages users to replicate the system and execute experiments. A 'System Replication' section provides instructions for Ubuntu 14.04 (Trusty), including a list of packages to install. A note mentions that in the Bielefeld University/TechFak environment, these packages are not needed. The instructions include a terminal command to install various development libraries and tools. A final note explains the use of the distribution tool chain, mentioning 'Bootstrapping' and 'Installing' sections, and provides a terminal command to bootstrap the 'icub-nightly.distribution'.

CITK catalogue

> About > Contact Search...

CATALOGUE ▾ BROWSE ▾ SEARCH ▾ VIRTUAL ▾ CITK TUTORIALS ▾ FLIER'S CONTRIBUTIONS

Home / Systems / iCub Ball Tracking / Version /

## iCub Ball Tracking-nightly

View Edit Track Devel

**Version:** nightly  
**Acronym:** icub-btracking  
**Version of System:** [iCub Ball Tracking](#)  
**Videos:** [iCub Ball Tracking Demo Video](#)

**Description:** In this particular system version, iCub related tools (yarview, iCubGui and dataSetPlayer) are used replay a pre-recorded sequence as provided in the [iCub Wiki](#) in order to visualize the original state of the robot. Feel free to **1) replicate** the system and **2) execute** the linked experiment to verify our results. Please see Linked Artifacts section. For a first glimpse, you may also check the linked video above.

### System Replication

For now, we have tested the replication with Ubuntu 14.04 (Trusty). Please install the following packages first.

In the Bielefeld University/TechFak environment you don't need to install these packages

**Ubuntu 14.04 LTS**

```
sudo apt-get install libace-dev libgsl0-dev libc6 python libncurses5-dev libgtkmm-2.4-dev libgladem-2.4-dev libqt4-dev \
libqt4-opengl-dev libcv-dev libhighgui-dev libcvaux-dev libstdl1.2-dev git gfortran freeglut3-dev cmake libxmu-dev libode1 \
libode-dev libopencv-dev libblas-dev python-dev openjdk-7-jdk swig gnuplot eog unzip
```

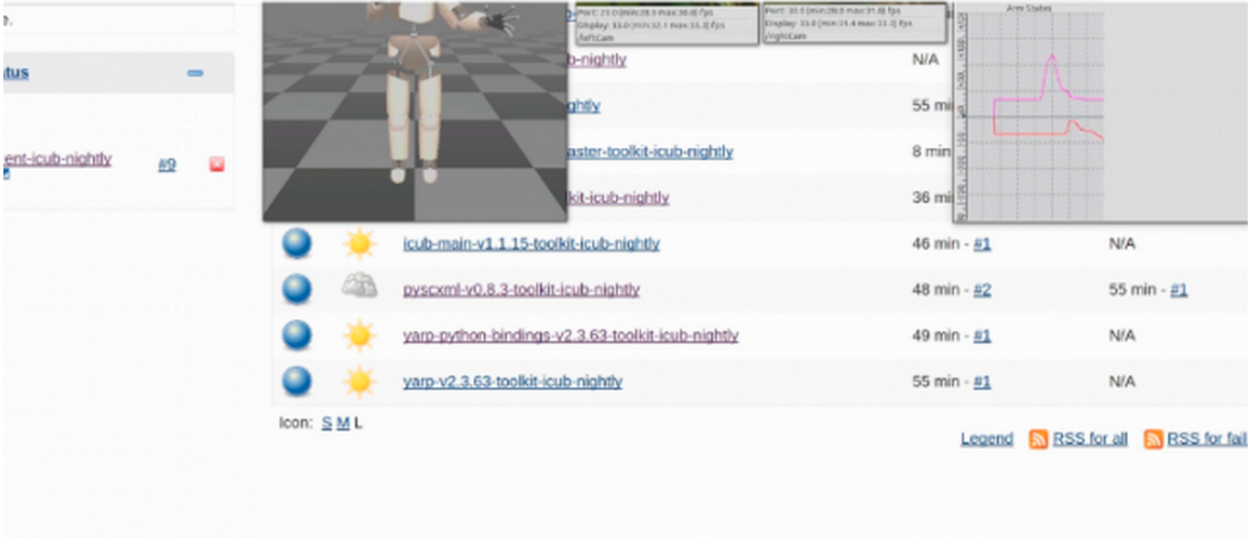
Now, please use our distribution tool chain as explained in the tutorials section [Bootstrapping](#) and [Installing](#). You may skip the MAVEN part as it is not required for this system version. Read and execute these instructions carefully. You will need to bootstrap the **icub-nightly.distribution**. If you changed your `$prefix`, please keep that in mind.

```
$prefix/jenkins/job-configurator --on-error=continue -b http://localhost:8080 -d $prefix/dist/distributions/icub-nightly.distribution -t '$prefix/
dist/templates/*.template'
```

Visit system version on catalog website



# Systems in Experimental Robotics | Step by Step



The screenshot displays a software interface for experimental robotics. It features a 3D model of a robot on a checkered floor. To the right, there is a table listing component versions and their associated metrics. Below the table, there are icons for 'GENERAL INFORMATION' and 'LINKED ARTIFACTS'. The 'LINKED ARTIFACTS' section lists various required components, linked distribution recipes, and linked experiments.

Component	Version	Time	Other
icub-main	v1.1.15	46 min - #1	N/A
pyscxml	v0.8.3	48 min - #2	55 min - #1
yarp-python-bindings	v2.3.63	49 min - #1	N/A
yarp	v2.3.63	55 min - #1	N/A

**Required Component Versions:**  
Finite State Machine Based Testing (FSMT)-0.16  
Data set iCub Ball Tracking-095402  
iCub Main-v1.1.15  
Yet Another Robot Platform (YARP)-v2.3.63  
iCub Contrib-master  
Yet Another Robot Platform (YARP) Python Bindings-v2.3.63  
FSMT Experiment iCub Ball Tracking-master  
PySCXML-v0.8.3

**Linked Distribution Recipe:**  
icub-nightly

**Linked Experiments:**  
iCub Ball Tracking-nightly

Browse linked artifacts: Publications, Component Versions, Data Sets, etc...

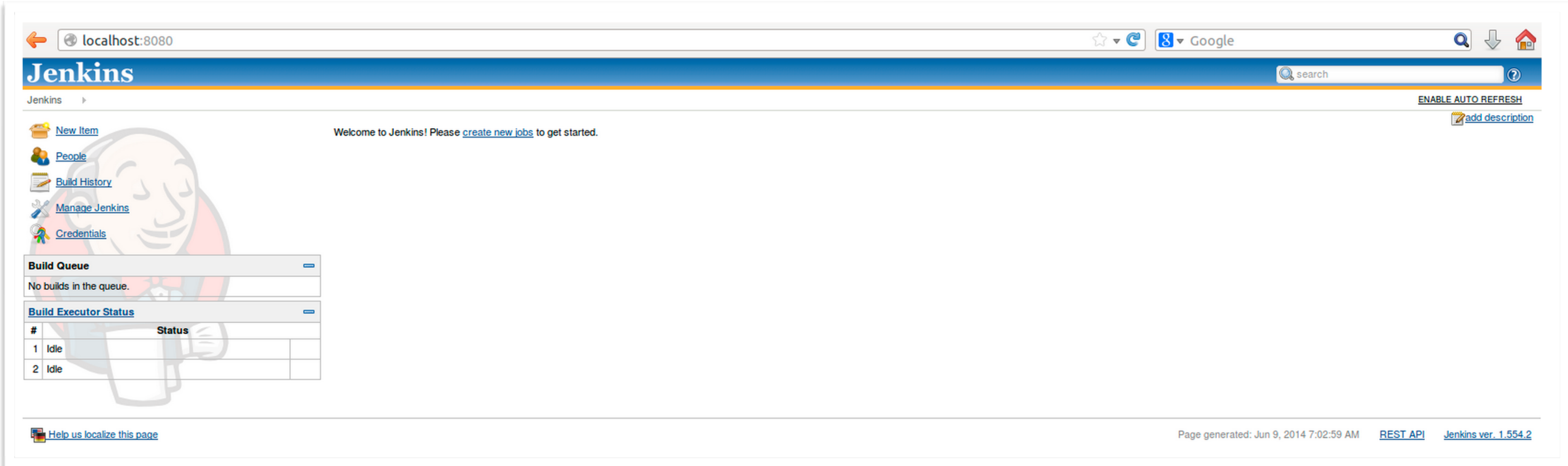
```
cd $prefix  
wget --no-check-certificate https://toolkit.cit-ec.uni-bielefeld.de/dist/jenkins.tar.gz
```

Download pre-packaged CI Server

```
cd $prefix/jenkins  
./start_jenkins
```

Extract & simply start it

# Systems in Experimental Robotics | Step by Step



Find ready to use local CI Server



```
cd $prefix  
mkdir dist  
cd dist  
git clone https://opensource.cit-ec.de/git/citk .
```

Clone CITK repository

# Systems in Experimental Robotics | **Step by Step**

```
/tmp/jenkins/job-configurator --on-error=continue -b http://localhost:8080 -d /tmp/dist/distributions/DESIRED_DISTRIBUTION.distribution -t '/tmp/dist/templates/*.template'
```

Invoke job-configurator with desired distribution

# Systems in Experimental Robotics | Step by Step

The screenshot shows the Jenkins dashboard interface. On the left, there are navigation links for 'New Item', 'People', 'Build History', 'Manage Jenkins', and 'Credentials'. Below these are sections for 'Build Queue' (showing 'No builds in the queue.') and 'Build Executor Status' (showing 4 idle executors). The main area displays a table of build jobs with the following data:

S	W	Name ↓	Last Success	Last Failure	Last Duration
🟡	☁️	<a href="#">dataset-icub-balltracking-095402-toolkit-icub-nightly</a>	1 day 10 hr - <a href="#">#5</a>	1 day 10 hr - <a href="#">#4</a>	24 sec
🟢	☀️	<a href="#">distribution-buildflow.icub-nightly</a>	1 day 11 hr - <a href="#">#3</a>	N/A	19 min
🟢	☀️	<a href="#">distribution-finish.icub-nightly</a>	1 day 10 hr - <a href="#">#3</a>	N/A	21 ms
🟢	☀️	<a href="#">distribution-prepare.icub-nightly</a>	1 day 11 hr - <a href="#">#3</a>	N/A	33 ms
🟢	☀️	<a href="#">execute-experiment-icub-nightly</a>	1 day 10 hr - <a href="#">#9</a>	N/A	1 min 43 sec
🟢	☀️	<a href="#">fsmt-0.16-toolkit-icub-nightly</a>	1 day 11 hr - <a href="#">#1</a>	N/A	3.3 sec
🟢	☀️	<a href="#">fsmt-exp-icub-nightly-master-toolkit-icub-nightly</a>	1 day 10 hr - <a href="#">#6</a>	N/A	6 sec
🟢	☁️	<a href="#">icub-contrib-master-toolkit-icub-nightly</a>	1 day 10 hr - <a href="#">#4</a>	6 days 0 hr - <a href="#">#1</a>	9 sec
🟢	☀️	<a href="#">icub-main-v1.1.15-toolkit-icub-nightly</a>	1 day 10 hr - <a href="#">#1</a>	N/A	9 min 53 sec
🟢	☁️	<a href="#">pyscxml-v0.8.3-toolkit-icub-nightly</a>	1 day 10 hr - <a href="#">#2</a>	1 day 11 hr - <a href="#">#1</a>	21 sec
🟢	☀️	<a href="#">yarp-python-bindings-v2.3.63-toolkit-icub-nightly</a>	1 day 10 hr - <a href="#">#1</a>	N/A	3 min 8 sec
🟢	☀️	<a href="#">yarp-v2.3.63-toolkit-icub-nightly</a>	1 day 11 hr - <a href="#">#1</a>	N/A	5 min 31 sec

At the bottom of the table, there are links for 'Icon: S M L' and 'Legend' with RSS feeds for 'RSS for all', 'RSS for failures', and 'RSS for just latest builds'.

Start build-flow job, the rest is orchestrated automatically, system is fully built (repeatable via single click)

# Systems in Experimental Robotics | Step by Step

Home / Experiments /

## iCub Ball Tracking-nightly

View Edit Track Devel

**Linked System:** [iCub Ball Tracking-nightly](#)  
**Experiment Recipe:** [iCub Ball Tracking](#)  
**Reference Data:** [11-17\\_233947.zip](#)  
**Video:** [iCub Ball Tracking Experiment](#)

**Description:** In this experiment, iCub related tools (yarpview, iCubGui and dataSetPlayer) are used replay a pre-recorded sequence as provided in the iCub Wiki in order to visualize the original state of the robot.

If you are running this system in a Virtual Machine, you will probably experience performance problems.

### Experiment Sequence and Expected Outcome

In this FSMT experiment multiple software components are used to visualize the iCub robot state. Thus, **first** a yarpservice is started followed by iCubGui and the dataSetPlayer (in hidden mode). After that, the dataSetPlayer is called via Yarp RPC to load the according data set. As soon as the data set has been fully loaded by the player the left-eye-view (yarpview) and right-eye-view are started. Now, two components are started that store the joint values published by the dataSetPlayer in a log file. Lastly, a script is called that connects all Yarp components (yarp-connect), the dataSetPlayer starts replaying the previously loaded data (player-play-data) — yarpscope starts (cf. Figure 1). After 70 seconds all components are tidily shutdown by FSMT, the acquired logs are saved. The desired outcome of this experiment is that the reference data (which is also distributed with this experiment) matches the replayed data (cf. generated plots). Additionally, only if all health checks passed, FSMT will report successful execution.

```
graph TD
    subgraph initialise_test
        eventless[eventless]
    end
    subgraph run_test
        subgraph state_0
            trigger0[on yarpservice.execute_program.success]
        end
        subgraph state_0_1
            trigger1[on icubgui.execute_program.success]
        end
        subgraph state_0_2
            trigger2[on datasetplayer.execute_program.success]
        end
        subgraph state_0_3
            trigger3[on player-load-data.execute_program.success]
        end
        subgraph state_0_4
            trigger4[on sleep.execute_program.success]
        end
        subgraph state_0_5
            trigger5[on left-eye-view.execute_program.success]
        end
        subgraph state_0_6
            trigger6[on right-eye-view.execute_program.success]
        end
        subgraph state_0_7
            trigger7[on dump-left.execute_program.success]
        end
    end
    eventless --> trigger0
    trigger0 --> trigger1
    trigger1 --> trigger2
    trigger2 --> trigger3
    trigger3 --> trigger4
    trigger4 --> trigger5
    trigger5 --> trigger6
    trigger6 --> trigger7
```

external queue internal queue

Visit linked experiment in the web catalog and review reference data

## Systems in Experimental Robotics | **Step by Step**

```
export prefix=/vol/icub-nightly/  
export PATH=$prefix/bin/:$PATH  
export PYTHONPATH=$PYTHONPATH:$prefix/lib/python2.7/site-packages/  
fsmt $prefix/etc/fsmt-experiments/icub-nightly/icub-nightly-balltracking.sxml
```

Invoke specified command (listed in catalog) and the experiment is executed locally

# Demo



# Systems in Experimental Robotics | Static Demo

The screenshot shows the Jenkins dashboard interface. On the left, there is a sidebar with navigation options: New Item, People, Build History, Manage Jenkins, and Credentials. Below these are sections for 'Build Queue' (showing 'No builds in the queue.') and 'Build Executor Status' (showing 4 idle executors). The main area displays a table of build jobs. The table has columns for status (S), weather icon (W), Name, Last Success, Last Failure, and Last Duration. Below the table are icons for size (S, M, L) and RSS feeds for all, failures, and latest builds. The footer indicates the page was generated on Nov 16, 2014, at 11:13:47 PM, and mentions the REST API and Jenkins version 1.580.1.

S	W	Name ↓	Last Success	Last Failure	Last Duration
		<a href="#">dataset-icub-balltracking-095402-toolkit-icub-nightly</a>	35 min - #5	36 min - #4	24 sec
		<a href="#">distribution-buildflow.icub-nightly</a>	55 min - #3	N/A	19 min
		<a href="#">distribution-finish.icub-nightly</a>	35 min - #3	N/A	21 ms
		<a href="#">distribution-prepare.icub-nightly</a>	55 min - #3	N/A	33 ms
		<a href="#">execute-experiment-icub-nightly</a>	N/A	N/A	N/A
		<a href="#">fsmt-0.16-toolkit-icub-nightly</a>	54 min - #1	N/A	3.3 sec
		<a href="#">fsmt-exp-icub-nightly-master-toolkit-icub-nightly</a>	7 min 38 sec - #6	N/A	6 sec
		<a href="#">icub-contrib-master-toolkit-icub-nightly</a>	36 min - #4	4 days 14 hr - #1	9 sec
		<a href="#">icub-main-v1.1.15-toolkit-icub-nightly</a>	46 min - #1	N/A	9 min 53 sec
		<a href="#">pyscxml-v0.8.3-toolkit-icub-nightly</a>	47 min - #2	55 min - #1	21 sec
		<a href="#">yarp-python-bindings-v2.3.63-toolkit-icub-nightly</a>	49 min - #1	N/A	3 min 8 sec
		<a href="#">yarp-v2.3.63-toolkit-icub-nightly</a>	55 min - #1	N/A	5 min 31 sec

https://vimeo.com/112005754

# Systems in Experimental Robotics | **Conclusions**

Modeling of artifacts required for system replication and experiment **execution**

'Easy to use' (yet to be proven) system deployment strategy for local and 'global' use case

Inherent **CI** paradigm for software and experiment provenance

Targets software developers, experiment designers and interested researchers / reviewers

Enables early integration of experiment designers — R, Matlab **scripts in the loop**

Ideal for early testing with Simulation environments (as shown in the demo)

Browsable web catalog of semantically linked research artifacts



# Systems in Experimental Robotics | **Current Shortcomings**

We modeled **HARDWARE** but we still need to provide a working use case

Faster 'shipping' methods, i.e., for reviewers only, using VM images or Linux containers like Docker

Automated data annotation, i.e., post experiment data processing like annotation of video material not supported

Build/Execution in the cloud (sorry) currently not investigated

Support for more than 2 Linux flavors, Mac OSX, probably Windows too

... and probably much more

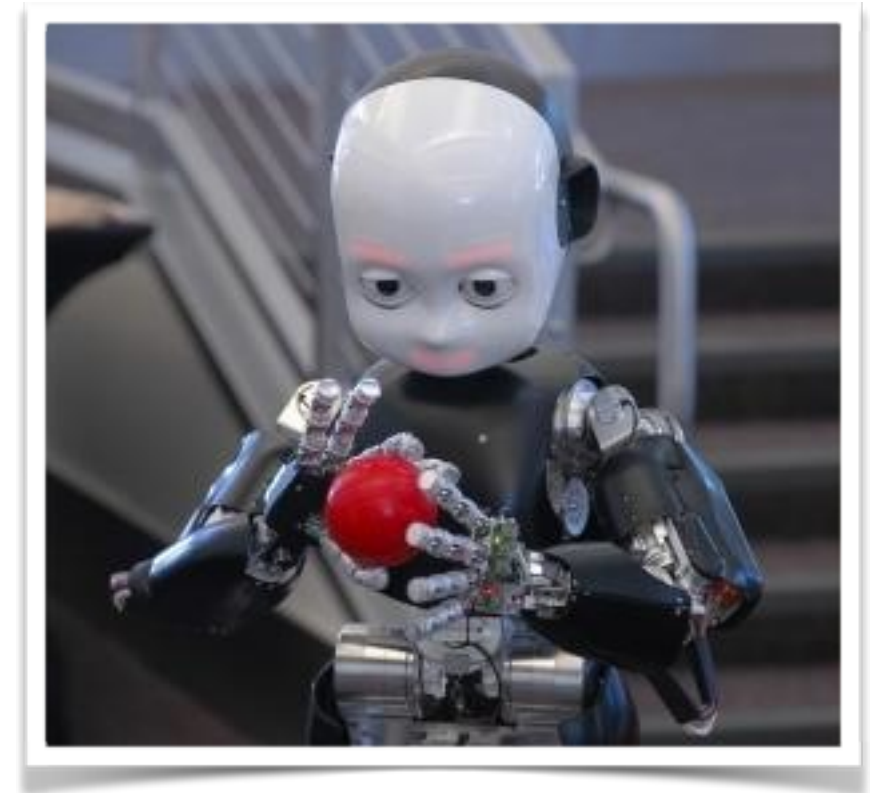
# Systems in Experimental Robotics | **Thank you!**

## Credits

Thanks Lorenzo for the invitation and Vadim for the technical support & the iCub friends for the **\_ready to use\_** software stack!



ISTITUTO ITALIANO  
DI TECNOLOGIA



# Systems in Experimental Robotics | **Thank you!**

## **Links**

Web Catalog: <https://toolkit.cit-ec.de>

Distribution Project: <https://opensource.cit-ec.de/citk>

Video Material: <http://vimeo.com/groups/citk>

## **Affiliation**

Florian Lier, Sven Wachsmuth and Sebastian Wrede are with Cognitive Interaction Technology — Center of Excellence, Bielefeld University ([www.cit-ec.de](http://www.cit-ec.de))

Sebastian Wrede is also with The Research Institute for Cognition and Robotics — CoR-Lab ([www.cor-lab.de](http://www.cor-lab.de))

## **References**

- [1] A. Bonarini et al. RAWSEEDS: Robotics advancement through web- publishing of sensorial and elaborated extensive data sets. In IROS'06 Work- shop on Benchmarks in Robotics Research, volume 6, 2006.
- [2] K. Dautenhahn. Methodology and themes of human-robot interaction: a growing research field. International Journal of Advanced Robotic Systems, 2007.
- [3] F. Bonsignorio, J. Hallam, and A. del Pobil. Defining the requisites of a replicable robotics experiment. In RSS2009 Workshop on Good Experimental Methodologies in Robotics, 2009.

Systems in Experimental Robotics | **Thank you!**

**HANDS ON SESSION — Let's integrate?**