Grant Agreement:	604102	Project Title:	Human Brain Project

Document Title:	Neurorobotics Platform v1
Document Filename:	SP10 D10.4.4 FINAL.docx
Deliverable Number:	D10.4.4
Deliverable Type:	Prototype
Work Package(s):	WPs 10.1 – 10.7
Dissemination Level:	PU
Planned Delivery Date:	M30 / 31 Mar 2016
Actual Delivery Date:	M30 / XX Mar 2016

Authors:	Axel VON ARNIM, FORTISS (P54), WP10.2.
Compiling Editors:	Axel VON ARNIM, FORTISS (P54), WP10.2
Contributors:	Axel VON ARNIM, FORTISS (P54), WP10.2, Marc-Oliver GEWALTIG (P1), WP10.1
Coordinator Review:	EPFL (P1): Jeff MULLER, Martin TELEFONT UHEI (P45): Sabine SCHNEIDER, Martina SCHMALHOLZ
Editorial Review:	EPFL (P1): Guy WILLIS, Lauren ORWIN, Colin McKINNON

Abstract:	This is the first public release of the Platform, available for use by any scientist, engineer or curious person. The Platform includes the Closed Loop Engine, World Simulation Engine, Web cockpit, Robot Designer, Environment Designer, Brain Interfaces & Body Integrator, Experiment Designer, Experiment Simulation Viewer components in their first version as well as a library of brain and robot models.

Keywords:	Platform Release, Neurorobotics Platform
Available at:	www.humanbrainproject.eu/ec-deliverables

Table of Contents

List of Figures and Tables

1. The Aim of this Document

This document provides access to the Neurorobotics Platform v1 and related information.

2. How to Access the Neurorobotics Platform

The Neurorobotics Platform is one of six ITC Platforms that comprise the HBP Scientific Research Infrastructure. All these Platforms can be accessed via the HBP Collaboratory web interface:

https://collab.humanbrainproject.eu

To access the Collaboratory, users need to get registered as HBP and Neurorobotics users. As explained in the section about User Adoption Strategy, we propose three user categories that give access to the platform with three levels of involvement. To request credentials, the user must fill in a form from the Platform home page.

http://www.neurorobotics.net/fileadmin/platform

This sets a registration request that has to be accepted by the SP10 Manager before they can get their credential per email. For the first public release of the Platform, only a limited set of users will be granted access, since our hardware resources are very limited.

Direct link to the Neurorobotics Platform on the Collaboratory:

https://collab.humanbrainproject.eu/#/collab/71/nav/405

3. Platform User Instructions

The Platform Documentation constitutes a separate Deliverable (D10.4.5 - Neurorobotics Platform v1 – Documentation), which includes direct links to Technical and User Documentation.

4. Platform Testing and Quality Strategy

4.1 Software quality and testing

SP10 has two levels of testing: unit (low-level) testing and integration (user level) testing. The first one is automated, using code-testing frameworks, well known in the software industry, such as Jasmine for frontend AngularJS code or nosetest for Python backend code. Every new line of code has to be covered with a unit test. In practice, we demand minimum of 95% of test coverage, to let some room for non-testable lines, which is a very high rate.

Code Coverages(Cobertura)

Job ↓	Packages	Files	Classes	Methods	Lines	Conditionals
neurorobotics.CLE #500	100%	100%	100%	N/A	87%	100%
neurorobotics.CLE.gerrit #1207	100%	100%	100%	N/A	87%	100%
neurorobotics.CLE.PyNN.0.8 #11	100%	100%	100%	N/A	87%	100%
neurorobotics.ExDBackend #455	100%	100%	100%	N/A	93%	100%
neurorobotics.ExDBackend.gerrit #1165	100%	100%	100%	N/A	94%	100%
neurorobotics.ExDFrontend #611	100%	100%	100%	95%	96%	87%
neurorobotics.ExDFrontend.gerrit #1685	100%	100%	100%	95%	96%	88%
neurorobotics.ExperimentControl #14	100%	100%	100%	N/A	91%	100%
neurorobotics.ExperimentControl.gerrit #38	100%	100%	100%	N/A	91%	100%
neurorobotics.nestify #46	100%	100%	100%	N/A	90%	100%
neurorobotics.WSLib.gerrit #4	100%	100%	100%	100%	100%	100%
Total:	100%	100%	100%	95%	91%	88%

Figure 1: Code coverage report on Jenkins

The second level is user testing. This one is for now exclusively done manually by testers that follow a test plan. A test plan is set up shortly before each release (every 3 months) and usually leads to 2 weeks of bug fixing. A test plan defines use cases and usage paths in the platform that allow the tester some freedom to test some unforeseen or unusual scenario. Our bug reporting form is available online here. The testers are usually development team members, but can also be managers or early users. In the future (SGA1), we will use an automated user testing system (already developed), that will automatically go over every user interface feature, just like a user would do. We will not drop manual testing, of course. Quality is very important in SP10 and generally in HBP. We follow industry standards like the use of a versioning tool (Git), unit testing, peer reviewing (Gerrit), continuous integration (Jenkins) and user testing. Some other SPs have the same workflow. In our Scrum project management methodology, we also put much effort into defining clear tasks (user stories) and concrete doneness and acceptance criteria (including quality constraints), so that finished work can safely be accepted as done with a common understanding.

The external libraries and software that we use, such as Nest or Gazebo are maintained using the same tools as our repositories. We have created repositories on our servers to not be dependent on external changes and update them when wanted from the official repositories. They are deployed using the same toolchain (Jenkins) as our own software.

4.2 Hardware resources and operation

Our hardware resources are detailed in Annex A. As explained there, they are limited and too many concurrent users will experience resource limitation. Our user adoption strategy should, in most cases, prevent that from happening. But in case it does, we provide the user with resource monitoring information in the Web Cockpit so that, if the user is not granted resources, they understand why.

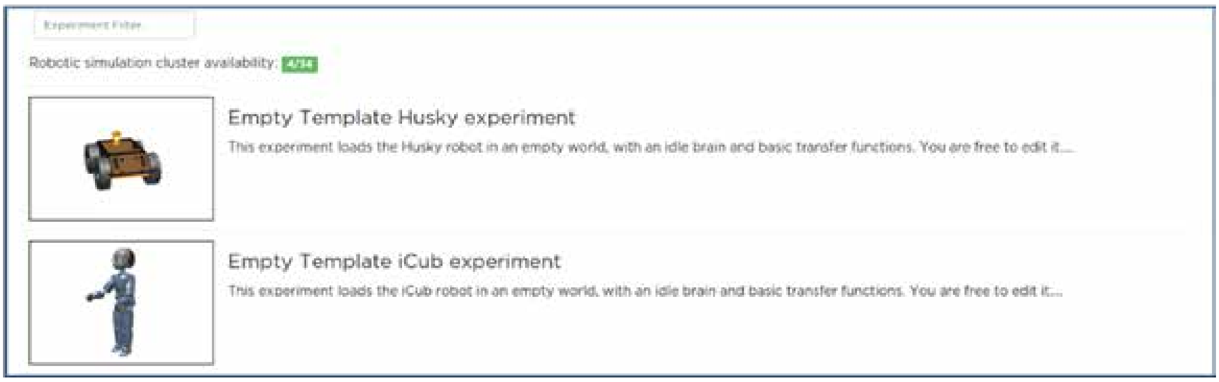

Figure 2: Cluster availability monitor

The maintenance of the hardware and infrastructure resources follows also strict quality rules. We manage all our servers with wide adopted tools like OpenStack and Puppet. We are able to spawn new servers rapidly, and we monitor operational failures continuously with Icinga.

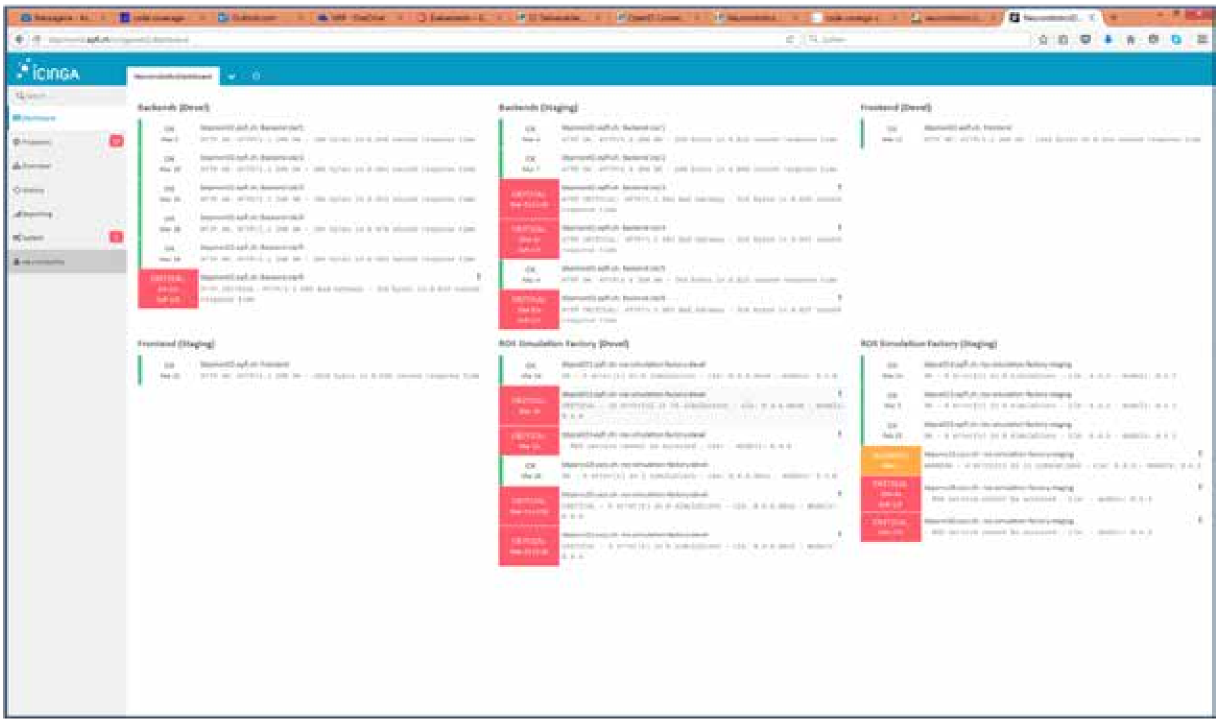

Figure 3: Icinga monitoring web interface

Operation is monitored also with Google Analytics. Thus, we learn about usage of the Platform and can adapt accordingly.

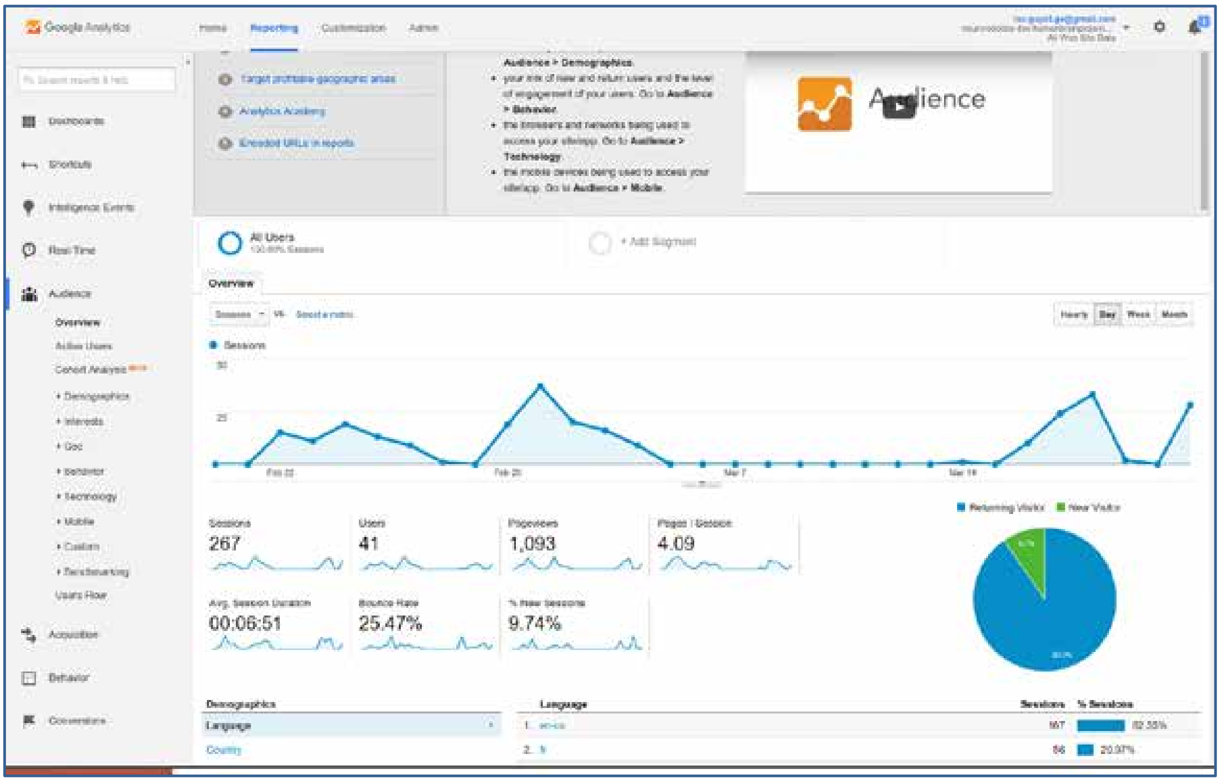

Figure 4: Google Analytics page showing usage of the Platform

4.3 Key performance indicators

Additionally to our software quality tools, we track other quality aspects using the Key Performance Indicators that we defined with the Scientific and Technical Coordinator. For example, we monitor development speed and outputs, backlog quality, peer reviewing quality. This monitoring happens after each sprint review (every third week) and is reported in the sprint report.

KPI	Title	Value(s)	Formula
SP10_sKPI_001	Code coverage by automatic tests	91%	
SP10_sKPI_002	No. of compiler warnings	0	
SP10_sKPI_003	No. of static analysis warnings	0	
SP10_sKPI_004	No. of user stories	504, 374	total, closed
SP10_sKPI_005	On-time delivery and backlog completeness	0-38/89 = -0,43	(No. of sprints until due date) - (remaining story points to complete/team velocity)
SP10_sKPI_006	Backlog estimation ratio	14/15 = 93%	No. estimated stories / no. stories (for the next version)
SP10_sKPI_007	Code reviewing efficiency	294/6358 = 46‰	Average no. of code review comments per 1000 lines of code
SP10_sKPI_008a	Speed of the Team 1	50/45 = 1,11	No. of points worked out in last sprint / mean speed of the team over last 3 sprints
SP10_sKPI_008b	Speed of the Team 2	96/44 = 2,18	No. of points worked out in last sprint / mean speed of the team over last 3 sprints
SP10_sKPI_009	Capacity to release fast	14/32 = 44%	No. of open bugs / estimated no. of bugs the two teams could solve in a dedicated bug solving sprint

Figure 5: KPI statistics in the Sprint report

5. Platform User Adoption Strategy

5.1 Target audience

The Platform is targeted at neuroscientists who wish to test brain models on robots and roboticists who want to test their robots with brain controllers. The first category is more academic while the latter can be also industrial.

5.2 Access policy

The whole Human Brain Project has a common approach towards user adoption. It defines a set of user categories: “basic HBP”, “Test” and “Project” with increasing access rights. Every user should request credential to access the Collaboratory Portal and our Platform. Though, an overview page with links to the documentation, a video tutorial, general description of the Platform is made publicly available, on our home page. From this overview page, users can request credential for any wanted category.

“Basic HBP” users get a permanent HBP account and access to the Neurorobotics Collaboratory, and can watch experiments run by others.

“Test” users get a renewable time slot for full access to the Platform. They have to detail in the request form why they want this type of account. They can test it, track its progress and report bugs. They also get online support.

“Project” users get full access for a long time period, dedicated resources and a liaison engineer for support. They have to give strong reasons why they want to work with us and detail their project in the request form. Typical first “Project” users will be CDPs.

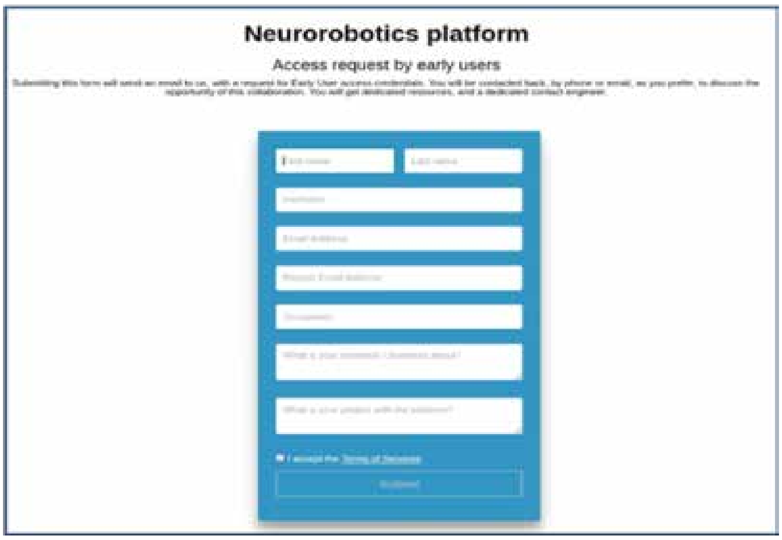

Figure 6: “Project” account request form on our public page

This not-so-open strategy is due to our lack of resources at this early stage of the Platform and to the fact that we want to control who gets full access on them. In later public releases, as we get more and more resources, this strategy will loosen.

5.3 User engagement

Users from “Community” or “Project” categories will be strongly linked to the Platform team, as they will get live support and personal relations with team members.

For all the others, though, tools like twitter, YouTube, our forum will keep them informed about our community actions. For example, for 2017 we would like to suggest two hands-on workshops:

- a course at the IK spring school: http://www.interdisciplinary-college.de
- a session at the European Robotics Forum: http://www.erf2016.eu

6. Help and User Feedback

Users will get help in many ways. First, all our documentation is available online from our home page at:

http://www.neurorobotics.net/fileadmin/platform

(see "Documentation" button at the very bottom).

For quick starters, we provide a video tutorial at:

http://www.neurorobotics.net/fileadmin/platform/videotutorial

We also provide support per email and chat for "Test" users and phone for "Project" users.

Our email is neurorobotics@humanbrainproject.eu

Our chat is available on the Neurorobotics collab at:

https://collab.humanbrainproject.eu/#/collab/71/nav/405

We also have a forum where users can find help, but also help others and share their experience.

https://forum.humanbrainproject.eu/

There is a support page in our collab that sums up all these links.

For user feedback, on top of the forum and email, users are invited to fill a survey form linked from the home page of the Platform. We also monitor usage with Google Analytics to learn about user habits and preferences.

6.1 User Feedback Received Month 18 – Month 30

Our users in the period M18-M30 were from SP11 – Applications and SP9 – Neuromorphic Computing. In WP11.1, they have been using the Platform to implement an experiment using a retina model developed by Greg FRANCIS and Eduardo ROS. We have provided technical support for the integration of their experiment into the Platform and they have provided models and controllers for the retina. The result is positive since the experiment is near to be released. As with other user cases, this experiment has driven many enhancements and features in our Platform.

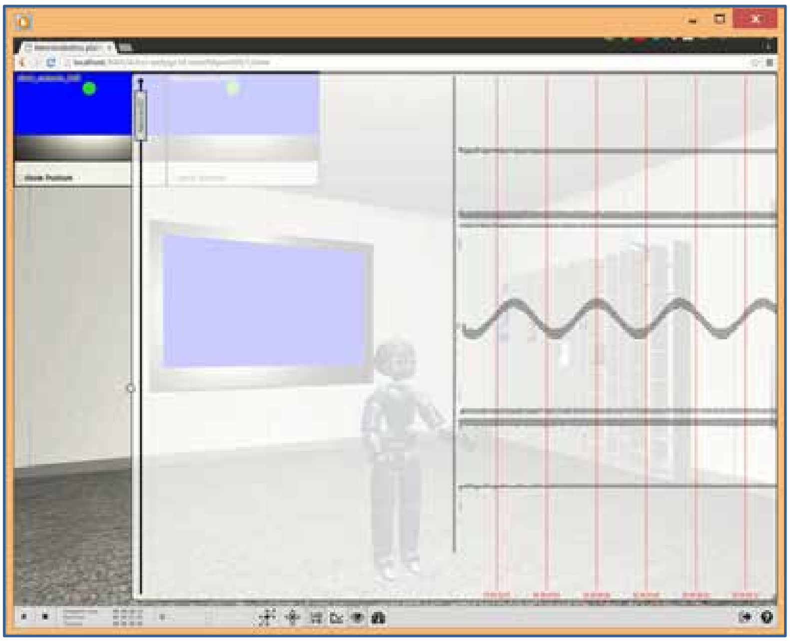

Figure 7: Retina experiment with iCub robot: plotting retina spikes (320 neurons)

Also part of WP11.1, researchers from FZI have integrated a Sensorimotor learning experiment using a virtual robotic arm that trains to move to unpredicted positions. The feedback was positive since the experiment was made possible.

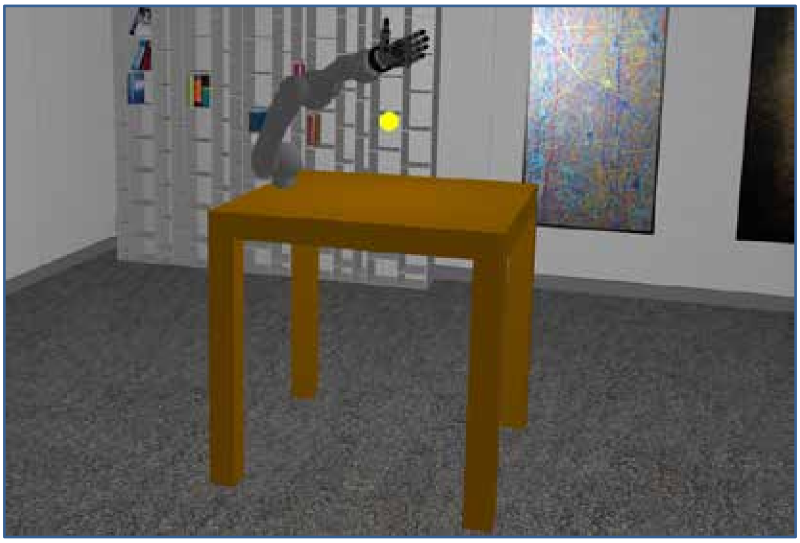

Figure 8: Sensorimotor learning experiment using Hollie Arm from FZI

Same, together with SP9 – Neuromorphic Computing, we have integrated our classical Husky-Braitenberg experiment running its brain model on a SpiNNaker board. We have provided the technical integration and they have provided the model and hardware. Proof of concept for brain integration was done by our SP9 users.

Figure 9: SpiNNaker board running the brain model of Husky-Braitenberg experiment

Annex A: Platform Architectural Diagram

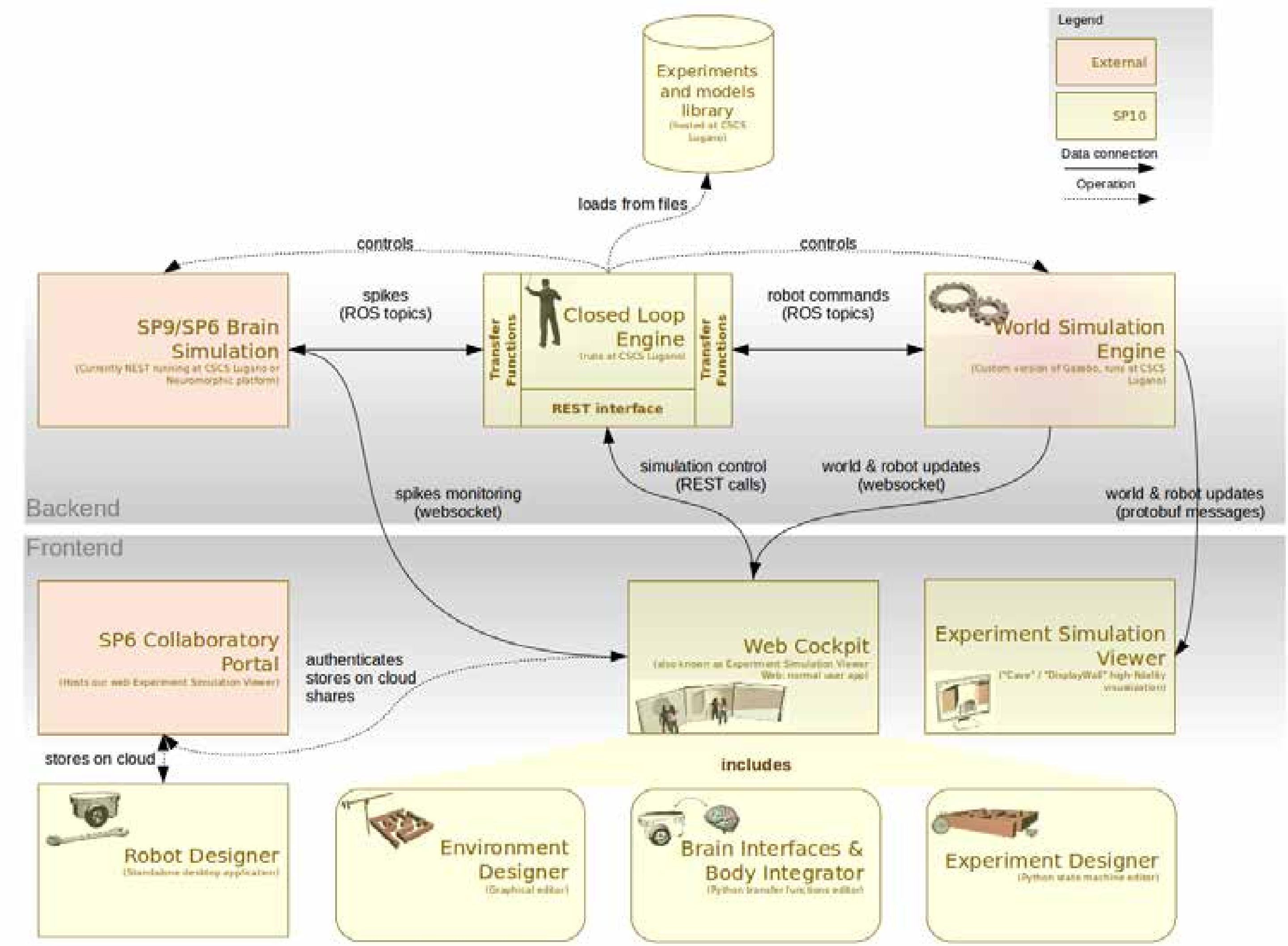

Figure 10: Platform architectural diagram

In the above diagram, SP10 and external components are identified by colour. Data flows are represented as full arrows and operations as dashed arrows.

From the hardware architecture viewpoint, apart from the DisplayWall, we use a set of virtual machines and cluster nodes at the CSCS centre in Lugano (CH). We have one virtual machine that runs the frontend server, and multiple virtual machines, called “backend” servers, that run the Closed Loop Engine and the Brain Simulation. These backend servers are all the same, they are replicated and managed by OpenStack and Puppet (well-known VM management systems). We can quickly spawn new ones, whenever we get new resources. For the public release, our resources are limited to 25 backend servers. One user running an experiment uses a full backend server. The World Simulation Engine runs on a cluster node that we allocate dynamically when a user starts an experiment. Opposite to backend servers, we do not have assigned cluster nodes, but we share them with other subprojects. So it is unpredictable if, at the time a user starts a simulation, they will get a node or not. This is of course monitored and explained to the user in case of failure.

During the Ramp-Up Phase, SP10 has developed mainly the vital backend Closed Loop Engine, World Simulation Engine components and the frontend Experiment Simulation Viewer. Designers are in a minimal version but cover most user needs, though not yet fully accessible to non expert users. The table below summarises the progress made on each component.

Table 1: Progress made on the SP10 components in the Ramp-Up Phase

Component	Product/Software Package/Service	Maturity level	Significant updates
Closed Loop Engine	Closed Loop Engine	High	Supports neuromorphic hardware, parallelised NEST
World Simulation Engine	World Simulation Engine	High	Supports more robot controllers, tactile feedback (used in SGA1)
Web Cockpit	Web Cockpit	High	Integrated in Collaboratory Portal, supports cloud storage and sharing, edit and save experiments, custom experiments, ...
Robot Designer	Robot Designer	Middle	First version. Supports export/import of neurorobotics models, kinematic chains.

Environment Designer	Web Cockpit	Middle	Basic environment editing, still, nice graphical UI, but still misses many features.
Brain Interfaces & Body Integrator	Web Cockpit	Low	Edition of transfer functions, edition of brain script and neuron populations. Misses graphical UI and brain visualisation.
Experiment Designer	Web Cockpit	Low	Basic script based event editor. Misses graphical UI.
Experiment Simulation Viewer	Experiment Simulation Viewer	Middle	No new feature since last review. Misses CAVE support and eventually better renderer.
Experiments and models library	Experiments and models library	Low	We have more experiments than in the last internal release but still too few for being called a "library"

Annex B: Software and Services Included in this Platform Release

Product/Software Package/Service name: Neurorobotics Closed Loop Engine

The closed loop engines synchronises the brain simulation and the World Simulation Engine. It provides a REST backend interface to control the simulation.

Category	Service
Tags	• backend • web • neurorobotics
Partners	• École Polytechnique Fédérale de Lausanne EPFL • Stiftung FZI Forschungszentrum Informatik am Karlsruher Institut fur Technologie FZI • Technische Universitaet München TUM • Scuola Superiore di Studi Universitari e di Perfezionamento Sant'Anna SSSA
Maintainers	• Axel von ARNIM (fortiss/TUM) • Luc GUYOT (EPFL) • Georg HINKEL (FZI)
Homepage	https://collab.humanbrainproject.eu/#/collab/71/nav/405
Documentation	https://collab.humanbrainproject.eu/#/collab/71/nav/1610
Support	https://collab.humanbrainproject.eu/#/collab/71/nav/4746
Source Code	git clone ssh://<user>@bbpcode.epfl.ch/neurorobotics/CLE, git clone ssh://<user>@bbpcode.epfl.ch/neurorobotics/ExDBackend, git clone

	ssh://<user>@bbpcode.epfl.ch/neurorobotics/GazeboRosPackages
License	**Missing License**
Current Version	1.0

Product/Software Package/Service name: Neurorobotics experiments and models library

Library of 3D robot and environment models. Library of template and example experiments.

Category	Library
Tags	• neurorobotics • 3d-models • simulation
Partners	• Stiftung FZI Forschungszentrum Informatik am Karlsruher Institut fur Technologie FZI • Technische Universitaet München TUM • École Polytechnique Fédérale de Lausanne EPFL • Scuola Superiore di Studi Universitari e di Perfezionamento Sant'Anna SSSA
Maintainers	• Axel von ARNIM (fortiss/TUM) • Luc GUYOT (EPFL)
Homepage	https://collab.humanbrainproject.eu/#/collab/71/nav/405
Documentation	https://collab.humanbrainproject.eu/#/collab/71/nav/1610

Support	https://collab.humanbrainproject.eu/#/collab/71/nav/4746
Source Code	git clone ssh://<user>@bbpcode.epfl.ch/neurorobotics/Models
License	**Missing License**
Current Version	1.0

Product/Software Package/Service name: Neurorobotics Experiment Simulation Viewer

High-fidelity rendering client application for use on DisplayWall, CAVE and desktops. It gives an immersive 3D representation of neurorobotics experiments with navigation capabilities.

Category	**Application**
Tags	• rendering • cave • neurorobotics • display-wall • 3d-models
Partners	• Technische Universitaet München TUM • École Polytechnique Fédérale de Lausanne EPFL
Maintainers	• Sandro WEBER (TUM) • Luc GUYOT (EPFL)
Support	https://collab.humanbrainproject.eu/#/collab/71/nav/4746
Source Code	git clone ssh://<user>@bbpcode.epfl.ch/neurorobotics/ESVRender

License	**Missing License**
Current Version	1.0

Product/Software Package/Service name: Neurorobotics Robot Designer

Blender plugin that enables to design and edit robot models and export and import them to and from the Neurorobotics platform.

Category	Application
Tags	• neurorobotics • simulation • robotics
Partners	• Stiftung FZI Forschungszentrum Informatik am Karlsruher Institut fur Technologie FZI
Maintainers	• Stefan Ulbrich (FZI)
Homepage	https://github.com/HBPNeurorobotics/BlenderRobotDesigner
Documentation	https://collab.humanbrainproject.eu/#/collab/71/nav/1610
Support	https://collab.humanbrainproject.eu/#/collab/71/nav/4746
Source Code	https://github.com/HBPNeurorobotics/BlenderRobotDesigner
Download Page	https://github.com/HBPNeurorobotics/BlenderRobotDesigner
License	**Missing License**
Current Version	1.0

Product/Software Package/Service name: Neurorobotics Web Cockpit

Neurorobotics web app. This is the main software package from SP10 provided to the users. It enables to create or edit neurorobotics experiments, and includes the Environment Designer, Brain Body & Body Integrator and Experiment Designer components.

Category	Application
Tags	• web • neurorobotics • simulation
Partners	• École Polytechnique Fédérale de Lausanne EPFL • Stiftung FZI Forschungszentrum Informatik am Karlsruher Institut fur Technologie FZI • Technische Universitaet München TUM • Scuola Superiore di Studi Universitari e di Perfezionamento Sant'Anna SSSA
Maintainers	• Axel von Arnim (fortiss/TUM) • Luc Guyot (EPFL)
Homepage	https://collab.humanbrainproject.eu/#/collab/71/nav/405
Documentation	https://collab.humanbrainproject.eu/#/collab/71/nav/1610
Support	https://collab.humanbrainproject.eu/#/collab/71/nav/4746
Source Code	git clone ssh://<user>@bbpcode.epfl.ch/neurorobotics/ExDFrontend
License	**Missing License**

Current Version	1.0

Product/Software Package/Service name: Neurorobotics World Simulation Engine

Forked version of Gazebo.

Category	Application
Tags	• gazebo • neurorobotics • simulation • robotics
Partners	• Stiftung FZI Forschungszentrum Informatik am Karlsruher Institut fur Technologie FZI • École Polytechnique Fédérale de Lausanne EPFL
Maintainers	• Fabian Aichele (FZI) • Luc Guyot (EPFL)
Homepage	http://gazebosim.org
Documentation	http://gazebosim.org/tutorials
Support	https://collab.humanbrainproject.eu/#/collab/71/nav/4746
Source Code	git clone ssh://<user>@bbpcode.epfl.ch/neurorobotics/gazebo
License	**Missing License**

Current Version	6.0.6.hbp.1.0

Annex C: Summary - Platform Use Case Status

Table 2: Platform use case status

Use Case ID	Description	Status/comments	TRL at end of Ramp-Up Phase	Related Product/Software Package/Service
SP10NRP-UC-001	Test a retina model on a brain model (offline simulation)	We do not support offline simulation in the ramp-up phase	1	All
SP10NRP-UC-002	Test a retina model on a brain model while interacting with the environment		1	All
SP10NRP-UC-003	Test bio-inspired learning algorithms (offline simulation)	We do not support offline simulation in the ramp-up phase	1	All
SP10NRP-UC-004	Test bio-inspired learning algorithms (offline simulation)	We do not support offline simulation in the ramp-up phase	1	All
SP10NRP-UC-005	Main use case: Assemble a complete virtual robot	Abigail can already load a robot from a common, well-established file format (compatible to the NRP) and Charlie can assemble a robot from individual parts to a certain extend.	2-3	Robot Designer
SP10NRP-UC-006	Assemble a virtual robot from robot parts	Not scheduled for the ramp-up phase. However, Individual kinematic chains can already be loaded and	2	Robot Designer

		assembled by Charlie (an experienced user). Geometric models can be imported from wide list of file formats and by specifying the kinematics (SP10NRP-UC-011) assemble a complete and functional robot model that can be used in the NRP or in other software of the robotics community.		
SP10NRP-UC-007	Load a virtual robot from robot library	A library access in the Robot Designer is not yet implemented.	1	Robot Designer
SP10NRP-UC-008	Import a virtual robot from file	Currently only local files.	4	Robot Designer
SP10NRP-UC-009	Customise robot's graphical properties	Charlie (the experienced user) can use the full potential of the underlying CAD software. Simplifications for Abigail are work in progress	2	Robot Designer
SP10NRP-UC-010	Attach sensors and actuators	Actuator models can be specified by Abigail. Sensor placement is work in progress.	3	Robot Designer
SP10NRP-UC-011	Define a kinematic chain	Charlie (the experienced user) can define a complete kinematics from scratch or edit individual	4	Robot Designer

		properties		
SP10NRP-UC-012	Customise robot's physical properties	Mass, centre of mass, inertia, maximal velocity, etc. can be specified by Charlie, the experienced user	3	Robot Designer
SP10NRP-UC-013	Add noise to sensors and actuators	Not implemented on the simulator side and interface has to be specified. Selection of noise models will be prepared until the end of the ramp-up phase	1	Robot Designer
SP10NRP-UC-014	Save robot	Robots can be saved as files for the underlying CAD software and be exported to (and be imported from) a well-established standard robot description format (URDF) and/or directly packaged for instant execution that can be processed the NRP and many components of the robotics community. Independent software libraries for reading and writing were implemented and made available in the repository.	3-4	Robot Designer
SP10NRP-UC-015	Change visualisation settings	This feature has been achieved directly by the choice of the underlying	5	Robot Designer

		CAD system. Additional short-cuts will be added to user interface and a dedicated section in the user manual will be added until the end of the ramp-up phase.		
SP10NRP-UC-015	Load and customise a robot from script	The Robot Designer is implemented as a plugin for a CAD software written in a scripting language and has a focus on reusable and well-defined building blocks. At the end of the ramp-up phase, there will be an extensive developer's manual available. Therefore, all of the functionality is available and can be accessed from a Python console or a built-in script editor - both provided and reliably implemented by the underlying CAD software. A dedicated short-cut will not be included in the user interface until the ramp up phase but the manual will guide an expert user with examples on how to use the Robot Designer for automated tasks and also how to extend the software to include	2	Robot Designer

		common automation task very easily for other users		
SP10NRP-UC-017	Assemble and model virtual environment	This use case is currently fully supported in the NRP	4	Web Cockpit World Simulation Engine
SP10NRP-UC-018	Define and save a building block	No multiple object selection available. No mechanisms for saving building blocks are currently implemented in the NRP	3	Web Cockpit World Simulation Engine
SP10NRP-UC-019	Load/import/export	Building blocks do not exist yet. A mechanism for the export is integrated but it is not available in the user interface.	3	Web Cockpit World Simulation Engine
SP10NRP-UC-020	Automatically generate a virtual environment from a script	This use case is supported in the NRP. An existing environment can be loaded from SDF by means of helper scripts.	4	Web Cockpit World Simulation Engine
SP10NRP-UC-021	Edit virtual environment	Only basic editing provided. Objects can be placed and moved, but some material properties cannot be set.	3	Web Cockpit World Simulation Engine
SP10NRP-UC-022	Define kinematic chain	This use case is currently unsupported in the NRP. Kinematic-chain based objects design will be implemented exploiting	2	Web Cockpit World Simulation Engine

		functionalities already available in the Robot Designer.		
SP10NRP-UC-023	Save virtual environment	This use case is fully supported in the NRP. The Virtual Environment can be saved in the Collaboratory Portal or in the user pc.	4	Web Cockpit World Simulation Engine
SP10NRP-UC-024	Visualise the virtual environment	This use case is currently fully supported in the NRP.	4	Web Cockpit World Simulation Engine
SP10NRP-UC-025	Selection of the brain model		1-2	Web Cockpit
SP10NRP-UC-026	Selection and grouping of neurons		1-2	Web Cockpit
SP10NRP-UC-027	Selection and grouping of neurons – query		1	Web Cockpit
SP10NRP-UC-028	Transfer modules creation		3	Web Cockpit Closed Loop Engine
SP10NRP-UC-029	Transfer modules connection – simple	Will be available shortly after public release	2	Web Cockpit Closed Loop Engine
SP10NRP-UC-030	Transfer modules connection – chain		3	Web Cockpit Closed Loop Engine
SP10NRP-UC-031	Transfer modules connection - direct feedback		3	Web Cockpit Closed Loop Engine
SP10NRP-UC-032	Transfer modules		2	Web Cockpit

	connection - environment connection – sensors			Closed Loop Engine
SP10NRP-UC-033	Selection of a configuration element		1	Web Cockpit Closed Loop Engine
SP10NRP-UC-034	Defining a measurement point		1	Web Cockpit Closed Loop Engine
SP10NRP-UC-035	Defining an action sequence	Programming events is possible by scripting, but not graphically	2	Web Cockpit Closed Loop Engine
SP10NRP-UC-036	Configuring a protocol		1	Web Cockpit Closed Loop Engine
SP10NRP-UC-037	Defining an experiment set-up		2	Web Cockpit Closed Loop Engine
SP10NRP-UC-038	Saving an experiment set-up		5	Web Cockpit Closed Loop Engine
SP10NRP-UC-039	Loading an experiment set-up		5	Web Cockpit Closed Loop Engine
SP10NRP-UC-040	Configuration and control via API	Fully implemented for interactive simulations	5	Web Cockpit Closed Loop Engine
SP10NRP-UC-041	Offline scenario with high fidelity visualisation		1	Experiment Simulation Viewer
SP10NRP-UC-042	Offline scenario without high fidelity visualisation		1	Experiment Simulation Viewer
SP10NRP-UC-043	Online scenario with a high fidelity visualisation	Display wall instead of cave	1	Experiment Simulation Viewer

				Web Cockpit
SP10NRP-UC-044	Developer scenario		5	Web Cockpit Closed Loop Engine
SP10NRP-UC-045	Loading an experiment set-up		5	Web Cockpit Closed Loop Engine
SP10NRP-UC-046	Configuring and starting the simulation	Selection of fidelity level is automatic, depending on the platform (web or Display wall)	4	Web Cockpit Closed Loop Engine
SP10NRP-UC-047	Time interaction	No support for step-by-step	3	Web Cockpit Closed Loop Engine
SP10NRP-UC-048	Reading neurons		1	Web Cockpit Closed Loop Engine
SP10NRP-UC-049	Exciting neurons		1	Web Cockpit Closed Loop Engine
SP10NRP-UC-050	Snapshotting the brain	Will be available shortly after public release	2	Web Cockpit Closed Loop Engine
SP10NRP-UC-051	Throwing artefacts at robot		1	Web Cockpit World Simulation Engine
SP10NRP-UC-052	Pushing the robot		1	Web Cockpit World Simulation Engine
SP10NRP-UC-053	Moving objects		5	Web Cockpit World Simulation Engine
SP10NRP-UC-054	Snapshotting the world	Can save the environment and robot position	5	Web Cockpit World Simulation Engine

SP10NRP-UC-055	Online simulation		1	
SP10NRP-UC-056	Offline analysis		1	

Annex D: Summary – Service IT Resource Planning

Table 3: Service IT resource planning

Product/Software Package/Service	TRL	Data Storage Capacity used by this Product	Data Storage Capacity Allocated for this Product	Location(s) of Data Storage	Data Access Protocol(s)*	Compute Resource(s) Allocated	Location(s) of Compute Resource(s) Allocated	Compute Access Protocol(s)**
Closed Loop Engine (25 instances)	4	25 x 1 Gb	25 x 11 Gb	CSCS Lugano	Local, Installed by Puppet	25 x 2 cpu/2Gb virtual machines	CSCS Lugano	Ssh, REST
Experiments and models library (25 instances)	3	25 x 270 Mb	25 x 11 Gb	CSCS Lugano	Local, Installed by Puppet	N/A	N/A	N/A
Robot Designer	3	< 100 Mb	User PC (Standalone desktop application)	User PC	Local (Installed by user from GitHub)	User PC	User PC	User PC
Web Cockpit (client)	4	< 200 Mb per user	1 Tb	Collab storage	Collab Python API	User browser	User browser	User browser
Web Cockpit (server)	4	13 Mb	11 Gb	CSCS Lugano	Local, installed by Puppet	1 x 2cpu/2Gb virtual machine	CSCS Lugano	Ssh, https
World Simulation Engine	4	1.23 Gb	10 Tb	CSCS Lugano	GPFS	1 cluster node dynamically allocated on Vizcluster	CSCS Lugano	Slurm
Experiment Simulation Viewer	4	< 100 Mb	11 Gb	Geneva DisplayWall	NFS4	DisplayWall cluster	Geneva DisplayWall	ssh

* Data Access Protocols such as GPFS, N.FS, S3, Collab storage, etc.

** Compute Access Protocols such as EC2, Task Framework, Unicore, OCCI, Slurm, ssh, gLite, Condor, etc.

Annex E: Backlog (Remaining bugs and new features to be added)

In this Annex we list the bugs and features that are related to the First Public Release, not beyond, and that are still open at the date of this writing. They are directly taken from our Jira backlog manager and are a bit raw to read, but the list gives a general idea of the status of the release.

Product/Software Package/Service name

Remaining Bugs

Key	Summary	Description
NRRPLT-3532	CLE should never return an empty dict	This causes errors in the frontend
NRRPLT-3531	CSV logging should be enabled by default	Add logging TF to every user-space experiment. Test it Change CSV button with shorter text
NRRPLT-3530	Mouse model should be rescaled (0.15) with all collision hulls	Exportes as URDF, SDF conversion works
NRRPLT-3514	Add light changes light position when experiment starts	Add a cylinder, Add a light, so that it lights the cylinder up. Click on play -> the lamp stays, but the light disappears
NRRPLT-3505	Reporting of TF Syntax Errors (Compile) doesn't work anymore	Open the Transfer Function editor and breaks the python indentation of some transfer functions. Then no error message is displayed in the TF editor (although the error is logged by the back-end). Note: the return of the failing PUT request ("apply" button of the TF editor) is purposely ignored as the error message is supposed to be passed through a ROS topic.
NRRPLT-3497	Internal error when closing	Axel Empty Husky template see bug2.png
NRRPLT-3491	Floating panes should have a close button	
NRRPLT-	Add object -> clicking on thumbnails	Axel

3486	selects page content and confuses user	
NRRPLT-3485	Add light -> when releasing mouse, light moves a bit	Axel: the light translates a little bit from the position you held the mouse.
NRRPLT-3484	Should be no minimum size for floating editor panel	Axel
NRRPLT-3481	transforming the lamp doesn't move the lamp spot light	Sandro: edit experiment lights are not treated as children of their respective models but as separate entities (e.g. transforming the lamp doesn't move the lamp spot light with it)
NRRPLT-3480	object selection in inspector often only works on second click	
NRRPLT-3479	positioning of lights error: after loading all lights are at pos (0,0,0). (sporadic error)	Sandro
NRRPLT-3477	Redirect run page to experiment list page when no experiment was clone	Sandro: add neurorobotics (dev) app into collab without cloning an experiment by clicking somewhere else in navigation when list of experiments is displayed selecting "(empty) my experiment title" in navigation leads to experiment list with 1 empty title "No experiment selected. Please click on the Edit button", both edit and launch buttons greyed (no available servers) and no way of cloning an experiment afterwards -> redirect to experiment list page
NRRPLT-3476	Changing the default experiment title causes errors	From Sandro: add neurorobotics (dev) app into collab without cloning an experiment by clicking somewhere else in navigation when list of experiments is displayed selecting "(empty) my experiment title" in navigation leads to a flood of error popups then immediately to the OIDC login screen (does not happen every time, error during setup of navigation/app entry?)
NRRPLT-3474	Trying to save to collab while timeout is running out results in failure	

NRRPLT-3472	Toolbar buttons icons (play/stop) show up as strange characters like "ä"	Reported by external user on Linux/firefox.
NRRPLT-3471	Splash screen frozen at asset loaded 0/0	This one is really annoying Debug or find a way to circumvent OR write on splash screen itself that in the case it is frozen, user should reload and join his own simulation. OR after timeout, close splash and automatically join experiment
NRRPLT-3460	cloned experiment shows other running instances of original template	CLone an experiment open your app in run mode run the original experiment in another tab from the NRP portal go back to your app you should see the other simualtion instance and be able to join, though you have a private copy of the exp if you change completely the experiment, you will still see other instances from the original exp you clones -> not good solution -> filter using contextID?
NRRPLT-3451	The great return of : Another simulation is already running on the server.	Steps to reproduce: When any error occur during the startup of an experiment, the server cannot create another one after.
NRRPLT-3450	Collision boxes are wrong for the plant and the lamp	Steps to reproduce: - Start a virtual room experiment - Open the object inspector for the virtual lamp and show the collision meshes - Same for the plant Issues: The plant has no collision meshes and the lamp collision is missplaced
NRRPLT-3420	Husky experiment (Collab): textures do not load 50% of the time	Load husky experiment from collab: 1/2 times the textures are not loaded and there is an error in the console log. If you click on the url of the error, you can open the experiment in separate tab and there the textures are loaded...
NRRPLT-3419	Dev iCub experiment (collab): load_h5_network() failure	Load icub experiment from dev servers:

		internal server error: load_h5_network() got unexpected keyword argument "sensors"
NRRPLT-3418	iCub experiment (collab): spikes not monitored	Start a collab icub experiment, open spikes editor.
NRRPLT-3414	Loading splash not showing up when simulation initialization is finished	Create a collab page, clone an experiment, edit it right after. sometimes, the loading hags on blue bar though "simulation initialized" is displayed.
NRRPLT-3412	Can't start Xvnc on the server, display port not available	Steps to reproduce: start a simulation on collab, happens 50% of the time while CLE is trying to start Xvnc on the server
NRRPLT-3392	Reading neurons' spikes from different TFs is not possible	When creating multiple Neuron to Robot TFs, spikes information is accessible only from the first one, the second and later ones will always receive 0 spike events and so behave as if the neurons didn't spike at all. This is likely due to PyNNNestSpikeRecorder.refresh() resetting the spike events and being called multiple times during a single simulation loop.
NRRPLT-3241	Remove temporary brain model from server on stop event	When using the NRP platform from the Collab, a copy of the brain model is uploaded to the CLE. This copy needs to be deleted when the simulation is stopped.
NRRPLT-3149	The Husky (VR) stops moving after a few seconds	This doesn't happen with Husky SBC (moon terrain). Is this due to the fact that Husky VR uses a different bibi config file with an external reference together with inlined python code?
NRRPLT-2914	Staging: ROS service cannot be accessed	tail -f /var/log/supervisor/roscore/roscore.err returns iterated errors of the following kind: run_id on parameter server does not match declared run_id: 28d87fbe-8162-11e5-aea4-fa163e63bfb9 vs 286ec4c0-8162-11e5-be5f-fa163e63bfb9 But also: Couldn't find an AF_INET address for [bbpsrvc29.cscs.ch] For this one, one may look at http://answers.ros.org/question/163556/how-to-solve-couldnt-find-an-af_inet-address-for-problem/ Besides, I noticed that bbpsrvc29 uses ROS indigo whereas bbpce025 uses ROS hydro.
NRRPLT-	Lauron slides and jumps with Gazebo 6	The Lauron robot behave strangly when ran with our Gazebo 6 modified version.

2855	from our repository	Steps to reproduce: - Start a Lauron experiment on a Lugano server - Wait for a few minutes You'll see the robot slightly jumping and also sliding on the floor. Maybe this is link to the bullet physics ?
NRRPLT-2723	The husky robot does not see the colors after an experiment reset	Step to reproduce: - Start an husky experiment on the Lugano cluster - Wait for the camera plugin to work (40 seconds) - Reset the experiment. - Click on play The robot turns round and never reach one of the red screen !
NRRPLT-2190	Lauron external controllers takes few minutes to be loaded	

Features

Key	Summary	Description
NRRPLT-3529	Easy download and install from GitHub	As a user of the RD; I want to download it from GitHub and install it easily. Acceptance: - installation instructions are clear - installation is as simple as possible - install goes flawless Doneness: - documentation updated in wiki as well
NRRPLT-3429	Add Husky with red balls experiment	
NRRPLT-3386	Change template SMACH code to change screen colour	As a user, I want to have a template code that actually works on default experiments and easy to understand examples. Acceptance: - default template code works in husky experiment

		- other commented out code snippets showcase other working events like light change, ... Doneness; Python: Code has been peer reviewed. Code coverage does not decrease. Code is built and packaged. Code is deployed on test servers. Code passes PEP8 and PyLint checks.
NRRPLT-3384	Neurorobotics app instance clones properly Husky and Lauron experiments +5	As a user I want all available user experiments to be cloned properly into the Collab space. Acceptance: - all user experiments are successfully cloned from the Collab and launched from the Collab Doneness: - Code has been peer reviewed. - Code coverage does not decrease. - Code is built and packaged. - Code is deployed on test servers. - Code passes PEP8 and PyLint checks.
NRRPLT-3343	Support URDF robot models in CLE	
NRRPLT-3324	Rebase to latest monsteer version.	* the code is synced with the latest stable monsteer release. * functionality tested after rebase. * tag in git repo to mark the version * code pushed to hbp-github

NRRPLT-3312	Remove developer page in staging version	As a user, I want not to access the developer experiments on the production servers. Acceptance: - config.json in puppet staging modified - link not shown on staging version - move server selector to user space Doneness: - puppet approved - Javascript / HTML / front-end : - Firefox, Chrome tested - Jenkins builds are green - The code is unit tested - The unit test code coverage does not decrease - The code is deployed on the dev server - The layout is tested for tablet screen sizes (768 px width according to bootstrap) - The code is peer reviewed - The manual is updated
NRRPLT-3266	Tutorial video for every component	As a user, I want to have very short videos showing how to use every component of the app. Acceptance: - small video for each designer, for navigation, for plots, .. - provide on youtube and on collab (as doc?) Doneness: - the videos are in the collab - the videos are on youtube - the videos are linked in the frontend (designers, help page) - the video have been reviewed by PO and Client
NRRPLT-3204	Interface between the retina and the brain +5 +8	As a user, I want to connect the retina to my brain. The retina have n*m output modules, each of them outputting a current (see retina/NEST_Module/COREM_Module/retina_interface.cpp)

		I want to be able to send these currents to some neurons of my brain. To this end, a custom device can be implemented (ask Georg for custom device). Acceptance criteria: - A retina device is implemented - The retina triggers activity in the brain - Multiple connection configurations are supported (one-to-one, all-to-all, ...) Doneness criteria: - Code is reviewed and merged - Jenkins is green - Unit tests are written - Deployed on dev servers
NRRPLT-3153	Documentation and testing	Doneness: * Unexperienced user can construct a simple robot * Documentation is helpful and guides her through the processes * Minimal Blender introduction * --Existing video is polished w/ nice voice and/or video cutting (help from Igor)--
NRRPLT-3098	ED: Reset world	As a user of the frontend, I want to be able to reset only models poses (not the robot) from the original SDF or their first place if they were added to the scene. Acceptance: - reactivate the functionality from gzweb Doneness: - repo - review - jenkins - tests (frontend) - firefox/chrome - tablet size
NRRPLT-3020	Run ESVRender on the DisplayWall + 5	As a user I want to watch my neurorobotics experiment on Geneva's DisplayWalls. Acceptance:

		- ESV render works with any "user" experiment - touchscreen navigation OK limited to zoom and translation Doneness: - repo - jenkins - unit tests - comments
NRRPLT-3005	Mouse experiment in virtual lab +3 + 0 + 2 + 2	AS a user of the mouse experiment, I want to have the y maze integrated into the fancy virtual lab, for the sake of esthetics and realism. Acceptance: - experiment shows y maze on table in virtual mouse - camera pose is such that the user can interact with the experiment directly - experiment works as usual Doneness: - repo - review - jenkins - comments - tablet size - coarse models - firefox / chrome
NRRPLT-2989	Move monsteer prototype to production	As a user of a brain, I want to have a reliable and performance efficient distributed brain in the NRP. Acceptance: - chosen solution is deployed / available in repos - user can inject spikes - user can get spikes - user can simulate t-time - MPI finishes cleanly and releases all nodes - simulation can be safely started/stopped - CLE is adapted

		Doneness: - code in repository - jenkins plan ok - reviewed - unit tested - code commented - documentation in wiki
NRRPLT-2876	Provide GUI list of available output / input neurons / populations +0 +8	As a user, I want to see which neurons / populations are avilable as inputs or outputs. Acceptance: Doneness: - repo - review - jenkins - comments - doc - tablet size - firefox / chrome

Annex F: IPR Status, Ownership and Innovation Potential

Table 4: IPR status, ownership and innovation potential

Product/Software Package/Service	IPR Status*	Owner(s)	Non-HPP users**	Innovation Potential***
Closed Loop Engine	Open Source	EPFL, TUM, Fortiss, SSSA		
Experiments and models library	Pre-IPR	EPFL, TUM, Fortiss, SSSA		May be sold if it gets bigger
Experiment Simulation Viewer	Open Source	EPFL, TUM, Fortiss, SSSA		Will evolve towards virtual CAVE, which would be a great innovation and have industrial potential
Robot Designer	Open Source	EPFL, TUM, Fortiss, SSSA		
Web Cockpit	Open Source	EPFL, TUM, Fortiss, SSSA		
World Simulation Engine	Open Source	Gazebo community		Additions to standard Gazebo are given back to the Gazebo community

* IPR Status: Open Source, Copyright, Patent, Trade Secret, pre-IPR (i.e. you intend to obtain some form of IPR in the future)

** If this product/software package/service is currently being used outside HBP (e.g. donated, loaned, licensed, sold), please specify by whom.

*** Innovation Potential: Potential practical applications beyond HBP, commercial and/or non-commercial.