Wireless Software and Hardware platforms for Flexible and Unified radio and network controL

Project Deliverable D1.2

IPR and Data Management Report

Contractual date of delivery:	31-08-2016
Actual date of delivery:	03-09-2016
Beneficiaries:	IMINDS, TCD, CNIT, TUB, NCENTRIC, RUTGERS, SNU
Lead beneficiary:	IMINDS
Authors:	Ingrid Moerman (IMINDS), Pieter Becue (iMinds), Alextian Liberato (TCD), Luiz DaSilva (TCD), Ilenia Tinnirello (CNIT), Pierluigi Gallo (CNIT), Piotr Gawlowicz (TUB), Robin Leblon (NCENTRIC), Ivan Seskar (RUTGERS), Changmok Yang (SNU)
Reviewers:	Spilios Giannoulis [IMINDS], Maicon Kist (TCD)
Work package:	WP1 – Project Management
Estimated person months:	2
Nature:	ORDP
Dissemination level:	CO
Version:	0.1

Abstract:

This internal deliverable stipulates the guidelines for IPR management within the WiSHFUL Project. This deliverable also includes the Data Management Plan.

Keywords:

IPR management, licensing models, access scenarios, Data Management Plan

Executive Summary

The first part of this document defines the **guidelines for IPR management o**f the software developed within the WiSHFUL Project and proposes a legal framework for access and use of the WiSHFUL software by External Parties during and beyond the Project.

First, a survey of the software assets and their owners is presented. A distinction is made between assets that already existed at the start of the project, referred to as 'basic software platforms', and assets that arise from the WiSHFUL Project. For the existing software platforms, we started from the initial legal ownership and Background, as stipulated by the Grant Agreement (GA) and refined in the Consortium Agreement (CA). For the software assets that arise from the WiSHFUL project, ownership of the Project Results is assigned as a function of the work generated in WiSHFUL.

Next, a summary is given of the initial legal framework for agreements between consortium partners about access rights dealing with the question how to offer Background and Results related to the WiSHFUL Project from one WiSHFUL partner to other WiSHFUL partner(s) in order to develop a practical joint solution ("access rights for implementation"). This initial legal framework also defines how access to Background and Results related to the WiSHFUL Project is given to External Parties ("access rights for exploitation of the Results"), that use the WiSHFUL software either as a Third Party in a funded Open Call, or a non-funded External Party.

Finally, this document proposes a more detailed legal framework for access and use of the WiSHFUL software by WiSHFUL partners, Third Parties or other External Parties, during or beyond the project, taking into account the specifics of the developed software during the course of the WiSHFUL project. Three scenarios for accessing the WiSHFUL software have been identified: (1) Access by Third Parties funded through WiSHFUL Open Calls during the WiSHFUL project; (2) Access by non-funded External Parties during and beyond the WiSHFUL Project; and (3) Access by WiSHFUL partners beyond the WiSHFUL Project. We further distinguish three access rules defining the conditions for access in terms of charges for using software, amount of technical support, and quotas for using experimental facilities: (1) guaranteed access, (2) best effort access and (3) premium access. In view of sustainable use of the WiSHFUL software, we support different licensing models for access to object code and source code for the three different use scenarios.

The second part of this document describes the ***Data Management Plan*** for the WiSHFUL project and follows the Guidelines on Data Management in Horizon 2020. It describes the data management life cycle for all research data sets to be collected, processed or generated by the project, and covers:

- the handling of research data during and after the project
- what data will be collected, processed or generated
- what methodology and standards will be applied
- whether data will be shared /made Open Access and how
- how data will be curated and preserved

The categories that require Open Access to be provided by the WiSHFUL partners, including organisations selected through its Open Calls, are:

- (public) Deliverables
- Conference/Workshop presentations (which may, or may not, be accompanied by papers, see below)
- Conference/Workshop papers and articles for specialist magazines
- Research Data and metadata

This Data Management Plan also provides Information about tools and instruments at the disposal of the beneficiaries and necessary for validating the results.

The document will evolve during the lifespan of the project, to take into account new software assets and corresponding IPR rules, new research data sets, any changes in consortium practices and external factors.

List of Acronyms and Abbreviations

AP	Access Point
BE-A	Best Effort Access
BG	Background
BSD	Berkeley Software Distribution
BSS	Basic Service Set
CA	Consortium Agreement
CLA	Contributor License Agreement
COAP	Constrained Application Protocol
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
DLA	Dual License Agreement
DMP	Data Management Plan
DOI	Digital Object Identifier
EDCA	Enhanced Distributed Channel Access
FPGA	Field Programmable Gate Array
G-A	Guaranteed Access
GA	Grant Agreement
GITAR	Generic Internet of Things ARchitecture
GPL	GNU General Public License
GPP	General Purpose Processor
GUI	Graphical User Interface
HTML	Hyper Text Markup Language
IoT	Internet of Things
IP	Intellectual Property
IPR	Intellectual Property Rights
IRIS	Implementing Radio In Software
jFed	Java-based framework for testbed federation
JSON	JavaScript Object Notation
LGPL	GNU Lesser General Public License
LPL	Low Power Listening
M2M	Machine-to-Machine
MAC	Medium Access Control
MEDCA	Moderated Enhanced Distributed Channel Access
OC	Object Code

OFDM	Orthogonal Frequency Division Multiplexing
OMF	Orbit Measurement and Management framework
OML	Orbit Measurement Library
OS	Operating System
P-A	Premium Access
SC	Source Code
SDR	Software Defined Radio
SQL	ructured Query Language
srs	Software radio systems
SW	Software
TAISC	Time-Annotated Instruction Set Computer
TBD	To Be Determined
TDMA	Time Division Multiple Access
TSCH	Time-Slotted Channel Hopping
UPI	Unified Programming Interface
US	United States
WARP	Wireless Open Access Research Platform
WiFi	Trademark for IEEE 802.11 standards
WiSHFUL	Wireless Software and Hardware platforms for Flexible and Unified radio and network controL
WMP	Wireless MAC Processor
XFSM	eXtended Finite State Machine

Table of contents

Appendix 1: Dual licensing agreement for commercial use and online academic use (DLA) 46

Appendix 2: Contributor License Agreement (CLA) 49

1 Introduction

The main focus of the WiSHFUL Project is to develop software modules with unified interfaces that permit wireless developers to quickly implement and validate advanced wireless network solutions in FIRE testbeds. This internal report stipulates the guidelines for IPR management of the software developed within the WiSHFUL Project. The main objective is to propose a legal framework for access and use of the WiSHFUL software by External Parties during and beyond the Project.

The WiSHFUL Project further chose to participate in the Open Research Data Pilot. This internal report therefore also includes the first version of the Data Management Plan describing the Project's policy and practices regarding the provision of Open Access to each of the public deliverables, presentations, scientific publications and research data sets that it will produce.

In order to improve the readability of the deliverable, a section with a non-exhaustive list of definitions of terms that are used in the deliverable is added at the beginning of the document.

2 Definitions

This section gives a non-exhaustive list of definitions of terms that are used in the deliverable:

Access rights	Rights to use results or Background under the terms and conditions laid down in the Grant Agreement and refined in the Consortium Agreement
Application Programming Interface	The application programming interface materials and related documentation containing all data and information to allow skilled Software developers to create Software interfaces that interface or interact with other specified Software.
Background	Any data, know-how or information — whatever its form or nature (tangible or intangible), including any rights such as intellectual property rights — that: (1) is held by the beneficiaries before they acceded to the Agreement, and (2) is needed to implement the action or exploit the results.
Controlled License Terms	Terms in any license that require that the use, copying, modification and/or distribution of Work and/or of Derivative Work be subject, in whole or in part, to one or more of the following: (1) that the Source Code or other formats preferred for modification be made available as of right to any third party on request, whether royalty-free or not; (2) that permission to create Derivative Work be granted to any third party; (3) that a royalty-free license relating to the Work or Derivative Work be granted to any third party.
Derivative Work	Any work that is a modified version of or is a derivative work of Work (see definition Work: 'Work' is software or another Work).
Exploitation	The direct or indirect utilization of *Results* in further research activities other than those covered by the *Project*, or for developing, creating and marketing a product or process, or for creating and providing a service, or for using them in standardization activities
External Party	A party having not signed the WiSHFUL Grant Agreement and the WiSHFUL Consortium Agreement or a subsequent Declaration of Accession.
Fair and reasonable conditions	Appropriate conditions, including possible financial terms or royalty-free conditions, taking into account the specific circumstances of the request for access, for example the actual or potential value of the Results or Background to which access is requested and/or the scope, duration or other characteristics of the exploitation envisaged. The beneficiaries must give each other access — under fair and reasonable conditions — to Background needed for exploiting their own results, unless the beneficiary that holds the Background has — before acceding to the Agreement — informed the other beneficiaries that access to its Background is subject to legal restrictions or limits, including those imposed by the rights of third parties (including personnel).
Third Parties	External Parties that are funded for performing experiments or implementing extensions as a result of a successful proposal in one of the "Open Calls". Third Parties are subcontracted by the Coordinator of the WiSHFUL Project.
Results	Any information or materials, whether or not they can be protected, which are generated under the Project. Such results include rights related to copyright; design rights; patent rights; plant variety rights; or similar forms of protection.

Object Code	Software in machine-readable, compiled and/or executable form including, but not limited to, byte code form and in form of machine-readable libraries used for linking procedures and functions to other software.
Open Call	During the term of the Project 5 Open Calls are organized to attract External Parties to use or extend WiSHFUL facilities and software platforms. The Open Calls are further detailed on the website http://www.wishful-project.eu/open-calls.
Project	The WiSHFUL Project.
Software documentation	Software information, being technical information used, or useful in, or relating to the design, development, use or maintenance of any version of a software program.
Software	Sequences of instructions to carry out a process in, or convertible into, a form executable by a computer and fixed in any tangible medium of expression.
Source code	Software in human readable form normally used to make modifications to it including, but not limited to, comments and procedural code such as job control language and scripts to control compilation and installation.
Uncontrolled License Terms	Any Software license that merely permits (but does not require any of) the things mentioned in (1) to (3) in the definition of Controlled License Term is not a Controlled License (and so is an Uncontrolled License).
Unified Programming Interface	A special Application Programming Interface developed in the WiSHFUL Project for monitoring and controlling wireless radios and networks
Work	Software or another work.

3 IPR Management

This section defines the guidelines for IPR management of the software developed within the WiSHFUL Project and proposes a legal framework for access and use of the WiSHFUL software by External Parties during and beyond the Project.

First, a survey of the existing software assets of and their owners is presented, further referred to as 'basic software platforms'. To this end we start from the initial legal ownership and Background, as stipulated by the Grant Agreement (GA) and refined in the Consortium Agreement (CA). Next, we identify the assets that arise from the WiSHFUL Project and assign ownership of the Project Results as a function of the work generated in WiSHFUL.

Next, a summary is given of the initial legal framework for agreements between consortium partners about access rights dealing with the question how to offer Background and Results related to the WiSHFUL Project from one WiSHFUL partner to other WiSHFUL partner(s) in order to develop a practical joint solution ("access rights for implementation"). This initial legal framework also defines how access to Background and Results related to the WiSHFUL Project is given to External Parties ("access rights for exploitation of the Results"), that use the WiSHFUL software either as a Third Party in a funded Open Call, or a non-funded External Party.

Finally, this document proposes a more detailed legal framework for access and use of the WiSHFUL software by WiSHFUL partners, Third Parties or other External Parties, during or beyond the project, taking into account the specifics of the developed software during the course of the WiSHFUL project. The legal framework includes (1) scenarios for accessing the WiSHFUL software, (2) access rules defining the conditions for access in terms of charges for using software, amount of technical support, and quotas for using experimental facilities, and (3) different licensing models for access to object code and source code.

3.1 Software assets

3.1.1 Basic software platforms

a. Wireless MAC Processor (WMP)

The Wireless MAC Processor (WMP) is an experimentation platform which offers the possibility to easily program, load and execute customized MAC protocols for WLAN, by using a platform-independent, high-level programming language, without scarifying the execution performance. This is usually not possible on common WLAN cards, because MAC protocol operations, such as backoff countdown, checksum verification, acknowledgment management, etc., are very time-critical and are implemented into the card hardware/firmware.

The basic idea of the WMP architecture is decoupling the elementary hardware events and actions required by a MAC protocol (the WMP application programming interface) from the logic according to which the functions are sequentially composed (the MAC program defined in terms of finite extended state machines) by a generic executor of protocols (the MAC engine). On top of this architecture, MAC protocols can be implemented and modified even more easily than in simulation. These achievements are protected by an US patent [1].

Two different prototypes of this architecture have been developed for two different hardware platforms: a commercial card by Broadcom and the SDR-based WARP research platform. For each prototype, different software modules are available: the WMP programming interface defined in the firmware and IP cores (for the WARP platform), the generic executor of MAC protocols implemented at the firmware level, a graphical editor and compiler for programming state machines and translating them in a machine-readable code, a control interface for loading the protocols inside the

platform. Because of the decoupling between hardware primitives and MAC logic, the same MAC protocol can be loaded on the two different hardware platforms.

The detailed description of the WMP software components is provided in Table 1.

Description of SW asset	Type of info and owner		Remarks
	Background	WiSHFUL Result	
WMP instruction set (core)	CNIT	CNIT	The initial set of events, conditions, and actions, which constitute the WMP instruction set was developed by CNIT in a previous project (EU FP7 FLAVIA). The instruction set has been enriched during the WiSHFUL project. The full list of the instructions has been released as open source for WiSHFUL partners, Third Parties and other external parties.
WMP instruction set (advanced)	CNIT	CNIT	Extra actions supporting advanced PHY capabilities have been developed by CNIT for the WARP platform within the WISHFUL project and the EU H2020 Flex5Gware project. The IP cores supporting these extra actions are not open source. The interface for using these actions is available for WiSHFUL partners, Third Parties and other external parties.
WMP execution engine for Broadcom cards	CNIT		The firmware implementing the WMP execution engine for a commercial card by Broadcom was developed during the EU FP7 FLAVIA project. The engine can be used by WiSHFUL partner, Third Parties and other external parties. The code is not open source.
WMP execution engine for WARP research boards	CNIT		The execution engine working on top of the WARP research platform was developed during the EU FP7 CREW project. The engine can be used by WiSHFUL partners, Third Parties and other external parties. The code is not open source.
Radio control monitoring & configuration extensions	CNIT	CNIT	A local controller for monitoring the low-level WMP signals and enforcing automatic adaptations was originally developed within the EU FP7 FLAVIA project. The controller has been extended for supporting the Local Control Services required by the WiSHFUL project, by extending opportunistically the WMP execution engine. The code is not open source.
XFSM graphical editor and compiler	CNIT	CNIT	The graphical editor allows MAC programmers to design their own protocol working on state-machine abstractions and neglecting hardware internals. A compiler allows converting the

			graphical representation of protocols into machine-readable code. The editor and the compiler have been developed within the EU FP7 FLAVIA project. They have been extended with new radio control monitor and configuration functions. The code is not open source.
MAC protocols (radio Programs): • CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) • TDMA (Time Division Multiple Access) • MEDCA (Moderated Backoff) • Multi-channel CSMA Cross-technology synchronization	CNIT	CNIT	The code for MAC protocols is developed by CNIT, and released as open source for WiSHFUL partners, Third Parties and other external parties.

Table 1: Software components of the WMP platform

b. Time-Annotated Instruction Set Computer (TAISC)

TAISC consist of a cross-platform MAC protocol compiler and execution engine. MAC protocols significantly impact wireless performance metrics such as throughput, energy consumption and reliability. The TAISC framework offers an abstraction for hardware independent MAC protocol design and management. The framework consists of a cross-platform MAC protocol compiler and an execution engine. The solution allows describing MAC protocols in a platform independent language (consisting of a radio platform independent instruction set), followed by a straightforward compilation step, yielding dedicated binary code, optimized for specific radio platforms. The cross-compilation approach allows developers to design MAC protocols once, and then compile them for reuse on different radio platforms. To enable time-critical operation, the TAISC compiler adds exact time annotations to every instruction of the optimized binary code. The execution engine running on the radio platform, will execute the instructions with accurate time control thanks to the time annotation.

The TAISC software platform is decomposed into different software components, as listed in Table 2.

Description of SW asset	**Type of info and owner**		**Remarks**
	Background	**WiSHFUL Result**	
TAISC instruction set	IMINDS	IMINDS	The initial instruction set is developed by IMINDS in previous projects, and released as open source for WiSHFUL partners, Third Parties and other External Parties. The instruction set may be enriched during the WiSHFUL project.
TAISC compiler	IMINDS		The compiler is developed by IMINDS in previous projects, and can be used by WiSHFUL

			partners, Third Parties and other External Parties. The code is not open source.
TAISC engine for embedded sensor platform RM090	IMINDS		The engine is originally developed by IMINDS in previous and parallel projects (IMINDS strategic IoT programme, SBO SAMURAI), and is being further improved and extended in the WiSHFUL project. The engine can be used by WiSHFUL partners, Third Parties and other External Parties. The code is not open source.
TAISC engine for embedded sensor platform Zolertia Re-Mote		IMINDS	As soon as available, the engine can be used by WiSHFUL partners, Third Parties and other External Parties. The code is not open source.
TAISC engine for Zynq based SDR platform		IMINDS	The development of the engine is also funded by other research Projects (a.o. H2020 Flex5Gware, IMINDS strategic IoT programme, SBO SAMURAI). As soon as available, the engine can be used by WiSHFUL partners, Third Parties and other External Parties. The code is not open source.
Radio control monitoring & configuration extensions		IMINDS	These extensions are integrated in the TAISC engine. The code is not open source.
MAC protocols (radio Programs): • CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) • Single channel TDMA (Time Division Multiple Access) • Channel hopping TDMA • LPL (Low Power Listening)		IMINDS	The code for MAC protocols is developed by IMINDS, and released as open source for WiSHFUL partners, Third Parties and other External Parties

Table 2: Software components of the TAISC platform

c. Generic Internet of Things ARchitecture (GITAR)

GITAR bundles control and management extensions for Internet-of-Things architectures running on constrained devices. The management extensions allow dynamic application and network level upgrades in an efficient way. GITAR enables automatically managing, updating and upgrading the software on IoT/M2M devices, particularly the network stacks, with new, improved functionality. The control extensions facilitate adding configuration and monitoring capabilities to network stacks in a generic manner. GITAR enables both local and remote configuration and monitoring of protocol parameters on constrained devices, allowing to fine-tune the behaviour and boost the performance even after deployment. The framework features a cross-layer design, which can be applied to any operating system running on IoT/M2M devices, and focuses on flexibility, allowing easy integration into existing network stack implementations with a minimal performance overhead (CPU, memory, energy). The management extensions of GITAR have been successfully implemented for Contiki OS The control extensions allow local configuration and monitoring via a native Contiki module and remote configuration and monitoring via a COAP interface.

The GITAR software platform is decomposed into different software components, as listed in Table 3.

Description of SW asset	Type of info and owner		Remarks
	Background	WiSHFUL Result	
GITAR engine for Contiki	IMINDS		The engine is originally developed by IMINDS in previous and parallel projects (IMINDS strategic IoT programme, SBO SAMURAI), and is being further improved and extended in the WiSHFUL project. The engine can be used by WiSHFUL partners, Third Parties and other External Parties. The code is not open source.
Monitoring and configuration extensions		IMINDS	Cross-layer adaptation for exposing configuration and monitoring options of existing network protocols. The code for these extensions is developed by IMINDS, and released as open source for WiSHFUL partners, Third Parties and other External Parties
Management extensions for runtime modular software updates	IMINDS	iMINDS	The management extensions are originally developed by IMINDS in previous and parallel projects (IMINDS strategic IoT programme, SBO SAMURAI), and are being further improved and extended in the WiSHFUL project. The management extensions can be used by WiSHFUL partners, Third Parties and other External Parties. The code is not open source.
Updated Contiki protocols		IMINDS	IMINDS has updated standard Contiki protocols for dynamic reconfiguration and monitoring. The code of these protocols is provided as open source for WiSHFUL partners, Third Parties and other External Parties
Modified Contiki build system	IMINDS		IMINDS has introduced a two-phase build process to automatically enable dynamic software updates. The code is not open source.

Table 3: Software components of the GITAR platform

d. Implementing Radio In Software (IRIS)

IRIS is a software architecture for building highly reconfigurable radio networks using a component-based design. The architecture is comprised of two categories: the IRIS core and IRIS modules. The IRIS core contains the core part of the IRIS framework, such as parsers, managers, and engines. The IRIS modules contain radio components for building software defined radios, including PHY-layer components and radio controllers. The IRIS architecture supports all layers of the network stack, providing a platform for the development of both reconfigurable point-to-point radio links and complete reconfigurable networks. To accomplish this, individual radios are described with an XML document, which specifies the components employed by each radio, the values for parameters, and connections between components. Components themselves are constructed as C++ signal processing blocks.

The IRIS software platform is decomposed into different software components, as listed in Table 4.

Description of SW asset	Type of info and owner		Remarks
	Background	WiSHFUL Result	
IRIS Core	TCD		IRIS Core is designed to be component-based. Discrete signal processing functions such as a digital filter or modulator are implemented as components with generic interfaces for lifecycle control, data passing, and reconfiguration. IRIS Core provides high flexibility, support higher layers of the network stack, and leverage the parallel processing capabilities of emerging processor platforms. IRIS is available to all external parties through an LGPL license.
IRIS reconfigurable random access MAC implementation	TCD		Implementation of carrier-sensing-based medium access control (MAC) protocols on inexpensive reconfigurable radio platforms has proven challenging due to long and un-predictable delays associated with both signal processing on a general purpose processor (GPP) and the interface between the RF front-end and the GPP. TCD has implemented a split-functionality architecture for a contention-based carrier-sensing MAC for IRIS, in which some of the functions reside on an FPGA (field programmable gate array) and others reside in the GPP. The code is not open source.
Cyclostationary signatures for softwared-defined radio	TCD		We define a *cyclostationary signature* as a feature, which may be intentionally embedded in a digital communications signal, detected through cyclostationary analysis and used as a unique identifier. We exploit cyclostationary signatures to overcome a number of the challenges associated with network coordination in emerging cognitive radio applications and spectrum sharing regimes. In particular we show their uses for signal detection, network identification and rendezvous in the context of dynamic spectrum access. We have implemented in IRIS a transceiver employing these techniques, using orthogonal frequency division multiplexing (OFDM). The code is not open source.
srsGUI (IRIS GUI)	TCD		srsGUI is a free and open-source graphics library for SDR using QT and QWT. The library provides a number of useful plots for graphing real and complex numbers. Each plot is designed to be update on-the-fly with the new

			data received by radio devices described using the IRIS Core.

Table 4: Software components of the IRIS platform

3.1.2 WiSHFUL control framework

The WiSHFUL Control Framework provides a simplified way for harmonized control of devices and protocols located on multiple wireless nodes. The architecture is divided into two categories: the core and modules. The core contains the main parts of the framework that allows for remote control and execution of UPIs. Three execution semantics are supported: synchronous blocking calls, asynchronous non-blocking calls and scheduled execution of UPI functions. The core provides mechanisms to experimenters allowing developing proactive and reactive control programs. Moreover, three types of control programs may be created: local, global and hierarchical. Behind a scene, core is responsible for node discovery, monitoring node presence, (de)serialization and transporting UPI and return values between controller and nodes. The core of WiSHFUL framework is being developed by TUB.

A WiSHFUL module is an entity that translates UPI into native device interface. In order to be controller, wireless device has to be associated with proper WiSHFUL module. Modules are developed by WiSHFUL project partners.

The framework is developed in Python programming language. The architecture is modular and code is split into Python packages, what ease development and maintenance.

The WiSHFUL Control Framework is decomposed into different software components, as listed in Table 5.

Description of SW asset	**Type of info and owner**		**Remarks**
	Background	**WiSHFUL Result**	
The Core of WiSHFUL Control Framework		TUB	The core of WiSHFUL Control Framework is developed by TUB and released as open source for WiSHFUL partners, Third Parties and other external parties. The core is developed and improved during the WiSHFUL project.
Node Discovery Module		TUB	TUB developed Node Discovery Module that allows all wireless nodes to automatically discover controller and connects to it. The code is open-source and can be used by WiSHFUL partners, Third Parties and other external parties.
Simple Module		TUB	The Simple Module is an example that may be used by novice users as starting point for their device modules. It shows how to use functionalities of WiSHFUL framework.

			The code is open-source and can be used by WiSHFUL partners, Third Parties and other external parties.
WiFi Device Module		TUB	The WiFi module is a wrapper for *iw* tool that is available in Linux OS. It provides functionalities as configuration of operating channel, setting transmission power, etc. The code is developed by TUB, and released as open source for WiSHFUL partners, Third Parties and other external parties.
Linux Networking Module		TUB	The Linux Networking module allows for configuration of iptables, Queueing Disciplines and routing tables. The code is developed by TUB, and released as open source for WiSHFUL partners, Third Parties and other external parties.
Mininet Integration		TUB	The WiSHFUL framework was integrated with WiFi-Mininet framework, what allows testing control programs in emulated environment. The code is developed by TUB, and released as open source for WiSHFUL partners, Third Parties and other external parties.

Table 5: Software components of the WiSHFUL control framework – Version 1.0

3.1.3 Platform-specific connectors to WiSHFUL control framework

a. WMP connector module

The WMP connector diverts the platform-independent programming interface offered by WiSHFUL into WMP-specific control commands and/or generic commands of Linux OS, such as iw, iwconfig, iptable.

Description of SW asset	**Type of info and owner**		**Remarks**
	Background	**WiSHFUL Result**	
WMP connector		CNIT	The code for these extensions is developed by CNIT and uses available libraries for the Broadcom platform, as well as the 802.11 reference design for the WARP platform. It is released as open source for WiSHFUL partners, Third Parties and other external parties.
WMP helper scripts	CNIT	CNIT	Helper scripts facilitate setting up APs, associating stations and configuring wireless interfaces. These are released as open source for WiSHFUL partners, Third Parties and other

			external parties.

Table 6: Software components of WMP connector module

b. GITAR/TAISC connector module for sensor nodes

The GITAR/TAISC connector module is responsible for transforming generic UPI calls from the WiSHFUL control framework into TAISC and GITAR specific control commands.

Description of SW asset	Type of info and owner		Remarks
	Background	WiSHFUL Result	
Transformation of generic UPI calls from WiSHFUL control framework to TAISC radio control and GITAR network control interface		IMINDS	The code for these extensions is developed by IMINDS, and released as open source for WiSHFUL partners, Third Parties and other External Parties.

Table 7: Software components of TAISC connector module

c. IRIS connector module

The IRIS connector module is responsible for transforming generic UPI calls from the WiSHFUL control framework in calls accepted by the IRIS controller. The connector module is a single piece of code with two communication open sockets. One socket accepts standard WiSHFUL UPI calls, while the other sends the translated UPI calls to the IRIS Controller.

Description of SW asset	Type of info and owner		Remarks
	Background	WiSHFUL Result	
Transformation of generic UPI calls from WiSHFUL control framework to messages accepted by the IRIS Core Controller		TCD	The code for these extensions is developed by TCD, and released as open source for WiSHFUL partners, Third Parties and other External Parties.

Table 8: Software components of IRIS connector module

d. Atheros connector module

ATH9K connector module implements UPI functions for Atheros wireless card. We exploit a fact that ATH9k Linux driver exposes non-standard functionalities of Atheros wireless card over *debugfs* and *Netlink* interface. ATH9K module inherits from WiFi module and adds following functionalities: configuration of EDCA parameters; configuration of transmission power per flow defined as 5-tuple; switching between TDMA and CSMA access technologies; spectrum scanning.

Description of SW asset	Type of info and owner		Remarks
	Background	WiSHFUL Result	
Ath9k Device Module		TUB	The code is developed by TUB, and released as open source for WiSHFUL partners, Third Parties and other external parties.

Table 9: Software components of Atheros connector module

e. LTE UE Connector Module (QMI based modem devices)

qmilteue connector module implements UPI functions for LTE devices based on Qualcomm chipsets from multiple vendors (e.g. Novatel, Huawei, Sierra Wireless, ZTE USB based wireless cards). These devices are using QMI (binary protocol designed to replace the AT command based communication with modems) with support for configuration and management of a number of services: Device Management Service (DMS), Network Access Service (NAS), Wireless Data Service (WDS), Location Service (PDS) and User Identity Module (UIM) service. The module currently implements basic connection management functionalities and statistics gathering.

Description of SW asset	Type of info and owner		Remarks
	Background	WiSHFUL Result	
qmilteue device module (tested with Netgear AC340/341U LTE UE and Sierra Wireless mc7455)		RUTGERS	The code is developed by RUTGERS, and released as open source for WiSHFUL partners, Third Parties and other external parties.

Table 10: Software components of LTE EUs connector module

3.1.4 Example Control Programs

A Control program is a piece of software that ***uses UPIs*** and ***implements the algorithm/logic*** that controls the radio and network protocol stack and adapts the behaviour of the wireless system to meet the QoS requirements defined by end users. Table 11 lists the Control Programs that have been developed by the WiSHFUL partners and offered to Third Parties and External Parties as examples for using UPIs in different use scenarios.

Description of SW asset	Type of info and owner: Background	Type of info and owner: WiSHFUL Result	Remarks
Simple Control Program		TUB	This is an example control program that shows how to use basic functionalities of WiSHFUL framework. It may be used by novice users as starting point for their control programs. The code is developed by TUB, and released as open source for WiSHFUL partners, Third Parties and other external parties.
EDCA Control Program		TUB	The example control program that shows how to configure EDCA parameters of WiFi devices. The code is developed by TUB, and released as open source for WiSHFUL partners, Third Parties and other external parties
Spectral Scan Control Program		TUB	The example control program that shows how to use spectral scan functionality provided by Atheros wireless card. The code is developed by TUB, and released as open source for WiSHFUL partners, Third Parties and other external parties
Hybrid MAC Control Program		TUB	The example control program that shows how to switch between CSMA and TDMA mode and configure its parameters. The code is developed by TUB, and released as open source for WiSHFUL partners, Third Parties and other external parties
Handover Control Program		TUB	The example control program that shows how to perform infrastructure-initiated handover in 802.11 networks. WiSHFUL UPIs are used to control also middle-boxes, ie. gateway and switches. The code is developed by TUB, and released as open source for WiSHFUL partners, Third Parties and other external parties
iptables Control Program		TUB	The example control program that shows how to program *iptables* of controlled nodes. The code is developed by TUB, and released as open source for WiSHFUL partners, Third Parties and other external parties
Control program for Changing MAC protocol	CNIT	CNIT	The example control program that shows how to active a specific radio program for WMP platform. The control program switch from CSMA (Carrier Sense Multiple Access) to TDMA (Time Division Multiple Access).

			The code is developed by CNIT, and released as open source for WiSHFUL partners, Third Parties and other external parties
Control program for automatic selection of MAC Protocol components (MetaMAC)	CNIT	CNIT	This control program implements a learning mechanism for selecting the best possible MAC protocol on the basis of the so called MetaMAC approach proposed in previous research work. Its adaptations to the WiSHFUL framework has been designed and implemented by CNIT. It is released as open source for WiSHFUL partners, Third Parties and other external parties.
Control program for MEDCA (Moderated Backoff)		CNIT	This control program implements a mechanism for selecting the contention window in CSMA protocols based on a recent proposal for 802.11 extensions by Qualcomm. The adaptation to the WiSHFUL framework and the implementation has been carried out by CNIT. It is released as open source for WiSHFUL partners, Third Parties and other external parties.
Control program for CW centralized tuning		CNIT	This control program is used in the showcase named “Load and Interference aware MAC adaptation”, and optimizes the nodes contentions windows according with the number of nodes present in the testbed. The code is developed by CNIT, and it is released as open source for WiSHFUL partners, Third Parties and other external parties.
Inter-BSS Wi-Fi synchronization		CNIT	This control program implements a mechanism to synchronize different BSS. It improves the network capacity in hidden nodes scenarios. The code is developed by CNIT, and it is released as open source for WiSHFUL partners, Third Parties and other external parties.
Cross-technology synchronization		CNIT/ iMINDS	Wi-Fi/ZigBee over-the-air synchronization.
Control program for changing MAC protocol in embedded sensor devices	IMINDS	IMINDS	The example control program is used in the showcase named “Load and Interference aware MAC adaptation”, and shows how to activate a specific radio program for TAISC platform. The control program switches from CSMA (Carrier Sense Multiple Access) to TDMA (Time Division Multiple Access). The code is developed by IMINDS, and released as open source for WiSHFUL partners, Third Parties and other external parties
Control program for CW centralized tuning of CSMA MAC on embedded sensor nodes		IMINDS	This control program is used in the showcase named “Load and Interference aware MAC adaptation”, and optimizes the nodes contentions windows according with the number of active traffic flows present in the

			testbed. The code is developed by IMINDS, and it is released as open source for WiSHFUL partners, Third Parties and other external parties.
Control Program for TSCH blacklisting		IMINDS	This control program is used in the showcase named “Coexistence of heterogeneous technologies”, and optimizes the operation of TSCH in wireless sensor network by blacklisting TSCH channels that collide with Wi-Fi channels The code is developed by IMINDS, and it is released as open source for WiSHFUL partners, Third Parties and other external parties.
Control program for STRALE (Standard-compliant & mobility-aware PHY rate & A-MPDU length adaptation)		SNU	This control program implements an algorithm which adapts PHY rate and A-MPDU length in the mobile environment. The degree of mobility is measured by checking subframe loss pattern in an A-MPDU. The code is developed by SNU, and released as open source for WiSHFUL partners, Third Parties and other external parties.

Table 11: Software components of Control Programsx

3.1.5 General Intelligence framework

The General Intelligence framework for enabling intelligent network and radio control is shown in Figure 1. The connection between the WiSHFUL software architecture for radio and network control and the intelligence framework is made by the Unified Programming Interfaces. As the UPIs are unified abstractions that span several wireless technology platforms, the components of the intelligence framework are generic. The Data Collection Component is a generic software module that interacts with the WiSHFUL UPIs, to retrieve data about radio and network operation, and with the Application API to retrieve information about the application requirements. The Data Collection Component also implements aggregation functionality to reduce the amount of data or to change the representation of data. The Intelligence Composition Module offers support for composing and configuring several algorithms available in the WiSHFUL Intelligence Repository into a self-contained intelligence engine that uses the data provided by the Data Collection Component and triggers configuration through the Action Component. The Action Component uses the WiSHFUL UPIs to adjust the configuration of radio and network. The WiSHFUL intelligence framework offers a common set of tools that enable the realization of intelligent approaches using the algorithms from the repository. Together with the UPIs the WiSHFUL software architecture of the intelligence framework enables reasoning about the current network state and applying actions to change the configuration of radio and network.

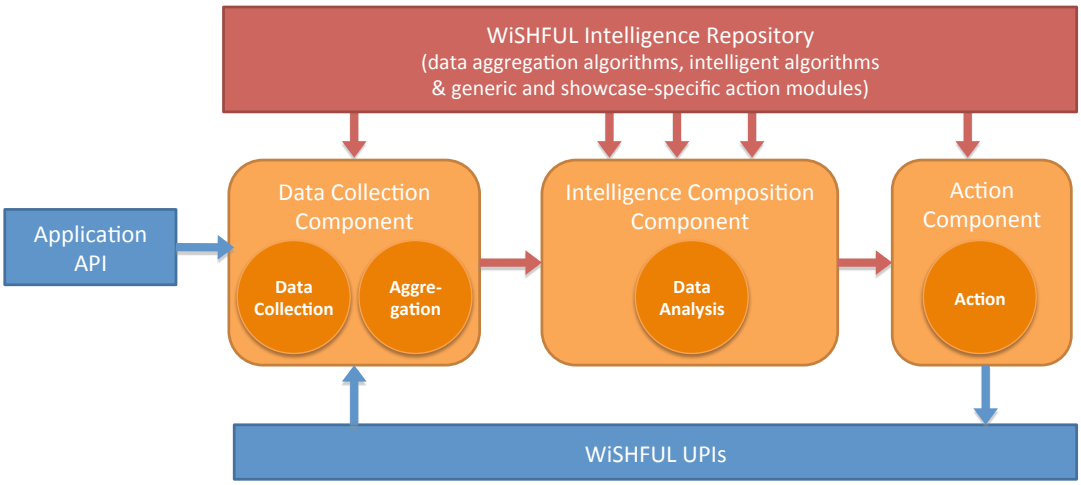

Figure 1: Conceptual view of the general Intelligence framework

The general Intelligence framework is decomposed into different software components, as listed in Table 12.

Description of SW asset	Type of info and owner		Remarks
	Background	WiSHFUL Result	
Node-RED, a tool originally developed by IBM, for wiring together hardware devices, APIs and online services (http://nodered.org)	IBM		Several tools have been analysed and Node-RED has been selected as the most promising open source tool for building the intelligence framework.
Extension of Node-RED with new services required for building intelligence workflows		IMINDS [more partners to follow]	The Intelligence framework is still under development and new services will be identified later in the project.

Table 12: Software components of the general Intelligence framework

3.1.6 Specific intelligence components

a. Data Collection component

The data component is responsible for data acquisition of the network status and the application requirements. With respect to the network status, the experimenter can specify the radio and/or network parameters he wants to monitor by choosing the parameters of interest from a predefined set of possible options (offered by the UPI interfaces) and the collection time window. With respect to the application requirements, a new interface (Application API) is needed to feed the application requirements to the intelligence framework.

The SW assets of the Data Collection component are listed in Table 13.

Description of SW asset	Type of info and owner		Remarks
	Background	WiSHFUL Result	
Control Programs that specify the data that needs to be collected and the storage of the data and according metadata.			Control Programs for Data Collection will be offered as open source to WiSHFUL partners, Third Parties and other External Parties.

Table 13: Software components for Data Collection

b. Data Aggregation algorithms

Through data aggregation algorithms collected data can be summarized or compressed to reduce the amount of data for enhancing network lifetime, to extract relevant features, or to change the representation of data.

The SW assets for Data Aggregation are listed in Table 14.

Description of SW asset	Type of info and owner		Remarks
	Background	WiSHFUL Result	
Minimum value of a series of data	TBD	TBD	Algorithms will be provided together with configuration options to WiSHFUL partners, Third Parties and other External Parties. Where possible, existing Implementations of such algorithms will be used (e.g SciPy [2] and Pandas [3]**).** In this case, necessary extensions/adaptors need to be implemented to allow the use of the algorithms from the (Node-RED) intelligence framework. More algorithms will be added and more information will be provided as soon as implementations of algorithms have been selected for integration in the WiSHFUL Intelligence framework. WiSHFUL will follow the same licensing approach as the original toolboxes. Most of these licenses are BSD compatible.
Average value of a series of data	TBD	TBD	
Median value of a series of data	TBD	TBD	
1^{st}, 3^{rd} quartile of a series of data	TBD	TBD	
Empirical probability and cumulative distribution functions	TBD	TBD	

Table 14: Software components for Data Aggregation

c. Intelligence Composition Component

The Intelligence Composition Component offers different approaches that can be selected by the experimenter for finding optimal radio and network settings. The intelligence modules will be offered as a collection of algorithms (e.g. optimisation and machine learning techniques) that can be applied for user-specific scenarios. The Intelligence Composition Component also offers modules for pre-processing data such as data cleaning (removing outliers), normalization, and data transformation.

Description of SW asset	Type of info and owner		Remarks
	Background	WiSHFUL Result	
Neural network	TBD	TBD	Existing implementations of machine learning algorithms will be used (such as for example the scikit-learn [4] tools or the Weka tools [5]). Some extensions may be developed by the WiSHFUL partners to integrate existing algorithms in the WiSHFUL Intelligence framework. WiSHFUL will follow the same licensing approach as the original toolboxes. Scikit tools are available under BSD license, while Weka tools use GPL
k-Nearest Neighbors algorithm	TBD	TBD	
Method to split the data into test and training	TBD	TBD	
Method to extract the confusion matrix for performance indication	TBD	TBD	
Prediction models that are the output of an offline learning phase (e.g. models that predict the performance of a MAC protocol based on density f nodes and interference level)		WiSHFUL partners (TBD)	Prediction models, developed by WiSHFUL partners using existing implementations of machine learning algorithms, will be provided as a black-box service to WiSHFUL partners, Third Parties and other External Parties
More algorithms will be added and more information will be provided as soon as implementations of algorithms have been selected for integration in the WiSHFUL Intelligence framework.	TBD	TBD	

Table 15: Software components for Intelligence Composition

d. Action Component

The Action Component represents an interface between the outputs of the intelligence algorithm and the UPI functions that enable the control of the behaviour of wireless nodes. This component translates the intelligence decisions taken by the Intelligence Composition Component in a sequence of UPI calls.

Description of SW asset	Type of info and owner		Remarks
	Background	WiSHFUL Result	
Intelligent Control Programs that use the intelligence model(s) from the intelligence toolbox online and generate UPI commands for configuration of radio and network settings			Intelligent Control programs will be offered as open source to WiSHFUL partners, Third Parties and other External Parties

Table 16: Software components for Action

3.1.7 Testbed Tools

a. FED4FIRE compliant testbed toolset

The FED4FIRE compliant testbed toolset contains several tools to assist the user in setting up and configuring advanced experiments on WiSHFUL testbeds. jFed provides easy selection and provisioning of resources, while Ansible allows the user to install complex software suites with lots of dependencies. In a next step, OMF6 can be used to do advanced experiment control and OML facilitates the collection of experiment results in a generic way.

The tools that are described in this section have been modified in WiSHFUL to include support for the TWIST and IRIS testbeds, as well as for the Portable Testbed. Next to this, some extensions were needed to these tools to allow easy integration of WiSHFUL UPIs in all testbeds. The different FED4FIRE compliant tools are listed in Table 18.

Description of SW asset	Type of info and owner		Remarks
	Background	WiSHFUL Result	
jFed	IMINDS	IMINDS	The jFed tool, originally developed by IMINDS in the FED4FIRE project, is being extended in the WISHFUL project to support all WiSHFUL testbeds. jFed is released under the MIT license.
Ansible	RED HAT	IMINDS	jFed has been extended with the ability to automatically start Ansible scripts. This automates the installation of all software required to support WISHFUL UPIs in the testbeds. Ansible is released by RED HAT inc. under the GNU GPL license. No modifications are made by WISHUFL to the Ansible software.
OMF6	NICTA	IMINDS	In WISHFUL, example OMF6 experiment

			description templates are made available to the community to ease the usage of WISHFUL UPIs in testbeds. OMF6 is released under the MIT license. No modifications are made by WISHFUL to the OMF6 software.
OML	NICTA	IMINDS	In WISHFUL, example OML configurations are made available to the community to ease the collection of experiment data formed by WISHFUL UPIs in testbeds. OML is released under the MIT license. No modifications are made by WISHFUL to the OML software.

Table 17: Software components of FED4FIRE compliant testbed tools

b. Portable testbed

The WiSHFUL project offers access to several wireless testbeds, such as TWIST (TUB), w-iLab.t (iMinds), IRIS (TCD), Orbit (Rutgers University) and a FIBRE Island at UFRJ. All of these testbeds are installed in either office environments or other dedicated testbed environments. Because some research requires doing measurement campaigns or actual testing in heterogeneous environments, the WiSHFUL project also offers a portable testbed to the community.

The Backbone Network Controller is an entity that controls behaviour and parameters of BN nodes using UPI functions. The current version of BN controller supports BN node discovery, integrity monitoring. Moreover, is uses UPI to install defined Queueing Disciplines in BN nodes, and to set operating channel for BN wireless network.

The tools that are used for testbed- and experiment management, and for controlling the Backbone Network are listed in Table 18.

Description of SW asset	**Type of info and owner**		**Remarks**
	Background	**WiSHFUL Result**	
Emulab	UTAH	IMINDS	Testbed management framework that exposes an SFA interface (to be used by jFed). Emulab is copyrighted by the Flux Research Group and the University of Utah. It is released under the GNU AGPLv3 license. Minor modifications were made to this framework to support all WISHFUL testbeds.
OMF6	NICTA	IMINDS	Control, measurement and management framework for testbeds. OMF6 is released under the MIT license. Minor modifications were made to this tool to support all WISHFUL testbeds.
OML	NICTA	IMINDS	Instrumentation tool for easy collection of measurement data. OML is released under the MIT license. Minor modifications were made to this tool to support all WISHFUL

			testbeds.
Backbone Network Control Program		TUB	The code is developed by TUB, and released as open source for WiSHFUL partners, Third Parties and other external parties
Wireless mesh solution for routing in backbone network	NCENTRIC	NCENTRIC	The code is developed by NCENTRIC, and will be provided as a black-box service to WiSHFUL partners, Third Parties and other External Parties using the portable testbed.

Table 18: Software components of the portable testbed

c. Generic data visualization tool

A new HTML5-based visualization tool allows the experimenter to convert measurement data into graphs in a generic way. The tool can be accessed through a browser and uses the SQL programming language to construct the graphs. The tool is able to visualize data originating from a postgreSQL or mySQL database. The tool is thus compatible with OML data sources.

Description of SW asset	**Type of info and owner**		**Remarks**
	Background	**WiSHFUL Result**	
PHP back end		IMINDS	The tool is developed in the WISHFUL project and allows generic visualization of data sources (incl. OML) using SQL queries. The PHP back end processes database queries and converts the data into JSON format. The visualization tool is released under the MIT license.
HTML5 front end		IMINDS	The front end converts the data from JSON format into uniformly looking HTML5 graphs. The visualization tool is released under the MIT license.

Table 19: Software components of generic data visualization tool

3.2 Legal framework for IPR management

3.2.1 IPR framework according to Grant Agreement and Consortium Agreement

The general IPR framework established by the Grant Agreement and Consortium Agreement is summarized in the Table 20.

	Rule	
	Background (BG)	**Project Results**
Sole ownership	Each partner is and remains sole owner	Each partner is and remains sole owner of own

	of independently generated BG. BG is clearly specified in the Consortium Agreement (CA). BG may be added during Project. Withdrawal of BG is only possible upon approval of the WiSHFUL steering committee BG can be excluded, but must be agreed by the Project partners.	generated Results If ownership is transferred, obligations are also transferred. Thirds Parties (in Open Calls) do not have any IPR rights on the Results of the Project.
Joint ownership	n.a.	Co-ownership takes place between partners that jointly generate joint Results. Joint owners jointly protect Results (and share costs for protection).
	Access rights for implementation	
Sole ownership	Access rights to BG are granted royalty-free and on a non-exclusive basis to the Project partners for the purpose of the Project. The requesting partner must show that the Access Rights are needed. Access rights can be conditional or restricted (limitations must be clear). Access rights do not include the right to grant sub-licenses.	Access rights are royalty-free to the Project partners for the purpose of the Project.
Joint ownership	n.a.	Access rights are royalty-free to the Project partners for the purpose of the Project.
	Access rights for exploitation	
Sole ownership	Access rights for using BG are granted under fair and reasonable conditions to other partners, if needed for generating own Results.	Access rights for using Results, are granted under fair and reasonable conditions to other partners, if needed to use own Results including for third-party research. Access rights for internal research activities will be granted on a royalty-free basis. A request for access rights may be made up to 12 months after the end of the Project, or, after the termination of the requesting partner's participation in the Project.
Joint ownership	Access rights for using BG are granted under fair and reasonable conditions to other partners, if needed to use own Results.	Each of the joint owners can use joint Results and grant non-exclusive licenses to third parties without the need for agreement or financial compensation from other joint owners. In case TUB and/or TCD is one of the joint owners a different agreement for the exploitation of the relevant joint Results will apply namely: (1) each of the joint owners shall be entitled to use their jointly owned Results for non-commercial research activities on a royalty-free basis, and without requiring the prior consent of the other joint owner(s), and (2) each of the joint owners shall be entitled to

		otherwise Exploit the jointly owned Results and to grant non-exclusive licenses to third parties (without any right to sub-license), if the other joint owners are given at least 45 calendar days advance notice, and Fair and Reasonable compensation.

Table 20: Summary of general IPR framework in WiSHFUL

3.2.2 Proposed IPR framework for access to WiSHFUL software

In this section, we propose a more detailed legal framework for accessing and using software generated in the WiSHFUL Project by External Parties or beyond the Project. We hereby mainly focus on the access rights and policies for use of the WiSHFUL software, as these are most important in view of access by External Parties (either funded in Open Calls or not funded through Open Access) and in view of sustainable operation beyond the WiSHFUL Project. Access rights for implementation are covered well by the Grant Agreement and the current consortium agreement (see previous section).

We distinguish three ***access scenarios*** for accessing the WiSHFUL software:

- Access by *Third Parties* funded through WiSHFUL Open Calls *during the WiSHFUL project*: This Third Parties are subcontracted to the coordinator of the WiSHFUL project and have to sign an agreement which stipulates the following terms with respect to IPR: “Copyright, other intellectual property and data protection legislations apply to software and data and subcontractors must respect them. The terms of applicable software and data licenses must be respected”.
- Access by *non-funded External Parties* during and beyond the WiSHFUL Project.
- Access by *WiSHFUL partners* beyond the WiSHFUL Project

We have to consider access to the WiSHFUL Project Results as well as to Background. Many of the new software tools developed within WiSHFUL build further on tools that were already available at the start of the Project (Background).

The general access rules to the Results and Background are clearly defined in the Grant Agreement and the Consortium Agreement (dated 4 August 2015). The general rule for accessing Results and Background states that access rights for use are granted under fair and reasonable conditions to other partners, if these Results and/or Background are needed to use own Results. In case of Results, access rights are also granted when own Results are used for third-party research. ‘Fair and reasonable conditions’ are not further defined in the Grant Agreement and the Consortium Agreement. In view of use of the WiSHFUL software by (funded) Third Parties during the WiSHFUL Project and External Parties during and beyond the WiSHFUL Project, these conditions needs to be very transparent in order not jeopardize the use of the software for the different access scenarios identified.

We distinguish the following ***access rules*** defining the conditions in terms of charges for using software, amount of technical support, and quotas for using experimental facilities:

- Guaranteed access (G-A): royalty free access and guaranteed support, guaranteed access to facilities
- Best effort access (BE-A): royalty free for non-commercial research, basic support, limited usage quota for access to facilities, no warranties
- Premium access (P-A): charging of (1) usage quota for guaranteed access to facilities and (2) staff for giving dedicated support

There may be further some ***restrictions on access***:

- Access limited to a certain physical testbed(s) or to a specific hardware configuration
- Need for agreement versus no need for agreement

As the Results and Background knowledge, created or used in the WiSHFUL Project are software tools, we should make a clear distinction between access rights to object code (OC) and access rights to source code (SC). Different licensing models for use of object code or access to source code can be followed. ***Licensing models*** adopted in WiSHFUL are:

- DLA: Dual licensing agreement for commercial use and online academic use (see Appendix 1), and according optional contributing license agreement (CLA) (see Appendix 2)
- IBM CLA: IBM International Business Machines, Inc. Contributor License Agreement [6]
- Apache 2.0 License [7]
- MIT: MIT License [8]
- GPL: GNU General Public License [9]
- LGPL: GNU Lesser General Public License [10]
- BSD: Berkeley Software Distribution [11]

An overview of the licensing models applied in WiSHFUL for the different software assets and access scenarios is given in Table 21. Appropriate measures are taken to protect the software made available on the WiSHFUL software repository.

SW asset	Access scenario		
	(Funded) Third Parties during Project	**(Non-funded) External Parties during/beyond the Project**	**WiSHFUL partners beyond the project**
BASIC SW PLATFORMS			
WMP			
WMP instruction set (core)	SC: DLA+CLA G-A	SC: DLA+CLA BE-A/P-A	SC: DLA+CLA BE-A/P-A
WMP instruction set (advanced)	SC: DLA+CLA G-A	SC: DLA+CLA BE-A/P-A	SC: DLA+CLA BE-A/P-A
WMP engine for Broadcom cards	OC:DLA G-A	OC:DLA BE-A/P-A	OC:DLA BE-A/P-A
WMP engine for WARP boards	OC:DLA G-A	OC:DLA BE-A/P-A	OC:DLA BE-A/P-A
WMP graphical editor and compiler	OC:DLA G-A	OC:DLA BE-A/P-A	OC:DLA BE-A/P-A
WMP radio control monitor and configuration extensions	OC:DLA G-A	OC:DLA BE-A/P-A	OC:DLA BE-A/P-A
WMP MAC protocols (radio programs)	SC: DLA+CLA G-A	SC: DLA+CLA BE-A/P-A	SC: DLA+CLA BE-A/P-A
TAISC			
TAISC instruction set	SC: DLA+CLA G-A	SC: DLA+CLA BE-A / P-A	SC: DLA+CLA BE-A / P-A
TAISC compiler	OC: DLA G-A	OC: DLA BE-A / P-A	OC: DLA BE-A / P-A

TAISC engine for embedded sensor platform RM090	OC: DLA G-A	OC: DLA BE-A / P-A	OC: DLA BE-A / P-A
TAISC engine for embedded sensor platform Zolertia Re-Mote	OC: DLA G-A	OC: DLA BE-A / P-A	OC: DLA BE-A / P-A
TAISC engine for Zynq based SDR platform	OC: DLA G-A	OC: DLA BE-A / P-A	OC: DLA BE-A / P-A
Radio control monitoring & configuration extensions	OC: DLA G-A	OC: DLA BE-A / P-A	OC: DLA BE-A / P-A
MAC protocols (radio Programs):	SC: DLA+CLA G-A	SC: DLA+CLA BE-A / P-A	SC: DLA+CLA BE-A / P-A
GITAR			
GITAR engine for Contiki	OC: DLA G-A	OC: DLA BE-A / P-A	OC: DLA BE-A / P-A
Monitoring and configuration extensions	SC: DLA+CLA G-A	SC: DLA+CLA BE-A / P-A	SC: DLA+CLA BE-A / P-A
Management extensions for runtime modular software updates	OC: DLA G-A	OC: DLA BE-A / P-A	OC: DLA BE-A / P-A
Updated Contiki protocols	SC: DLA + CLA G-A	SC: DLA + CLA BE-A / P-A	SC: DLA+CLA BE-A / P-A
Modified Contiki build system	OC: DLA G-A	OC: DLA BE-A / P-A	OC: DLA BE-A / P-A
IRIS			
IRIS Core	SC: LGPL G-A	SC: LGPL BE-A	SC: LGPL BE-A
IRIS reconfigurable random access MAC implementation	OC: DLA G-A	OC: DLA BE-A/P-A	OC: DLA BE-A/P-A
Cyclostationary signatures for softwared-defined radio	OC: DLA G-A	OC: DLA BE-A/P-A	OC: DLA BE-A/P-A
srsGUI (IRIS GUI)	SC: LGPL G-A	SC: LGPL BE-A	SC: LGPL BE-A
IRIS Core Controller Extension	SC: LGPL G-A	SC: LGPL BE-A	SC: LGPL BE-A
WiSHFUL CONTROL FRAMEWORK			
The Core of WiSHFUL Control Framework	SC: MIT G-A	SC: MIT BE-A	SC: MIT BE-A
Node Discovery Module	SC: MIT G-A	SC: MIT BE-A	SC: MIT BE-A
Simple Module	SC: MIT G-A	SC: MIT BE-A	SC: MIT BE-A
WiFi Device Module	SC: MIT G-A	SC: MIT BE-A	SC: MIT BE-A
Linux Networking Module	SC: MIT G-A	SC: MIT BE-A	SC: MIT BE-A

Mininet Integration	SC: MIT G-A	SC: MIT BE-A	SC: MIT BE-A
CONNECTORS TO WiSHFUL CONTROL FRAMEWORK			
WMP Connector Module	SC: DLA+CLA G-A	SC: DLA+CLA BE-A/P-A	SC: DLA+CLA BE-A/P-A
GITAR/TAISC connector module for sensor nodes	SC: DLA+CLA G-A	SC: DLA+CLA BE-A/P-A	SC: DLA+CLA BE-A/P-A
IRIS Connector Module	SC: LGPL G-A	SC: LGPL BE-A	SC: LGPL BE-A
Atheros Connector Module	SC: MIT G-A	SC: MIT BE-A	SC: MIT BE-A
LTE UE Connector Module	SC: BSD G-A	SC: BSD BE-A	SC: BSD BE-A
EXAMPLE CONTROL PROGRAMS			
Simple Control Program	SC: MIT G-A	SC: MIT BE-A	SC: MIT BE-A
EDCA Control Program	SC: MIT G-A	SC: MIT BE-A	SC: MIT BE-A
Spectral Scan Control Program	SC: MIT G-A	SC: MIT BE-A	SC: MIT BE-A
Hybrid MAC Control Program	SC: MIT G-A	SC: MIT BE-A	SC: MIT BE-A
Handover Control Program	SC: MIT G-A	SC: MIT BE-A	SC: MIT BE-A
iptables Control Program	SC: MIT G-A	SC: MIT BE-A	SC: MIT BE-A
Control program for Changing MAC protocol	SC: DLA G-A	SC: DLA BE-A/P-A	SC: DLA BE-A/P-A
Control program for automatic selection of MAC Protocol components (MetaMAC)	SC: DLA G-A	SC: DLA BE-A/P-A	SC: DLA BE-A/P-A
Control program for MEDCA (Moderated Backoff)	SC: DLA+CLA G-A	SC: DLA+CLA BE-A/P-A	SC: DLA+CLA BE-A/P-A
Control program for CW centralized tuning	SC: DLA+CLA G-A	SC: DLA+CLA BE-A/P-A	SC: DLA+CLA BE-A/P-A
Inter-BSS Wi-Fi synchronization	SC: DLA+CLA G-A	SC: DLA+CLA BE-A/P-A	SC: DLA+CLA BE-A/P-A
Cross-technology synchronization	SC: DLA+CLA G-A	SC: DLA+CLA BE-A/P-A	SC: DLA+CLA BE-A/P-A
Control program for changing MAC protocol in embedded sensor devices	SC: DLA+CLA G-A	SC: DLA+CLA BE-A/P-A	SC: DLA+CLA BE-A/P-A
Control program for CW centralized tuning of CSMA MAC on embedded sensor nodes	SC: DLA+CLA G-A	SC: DLA+CLA BE-A/P-A	SC: DLA+CLA BE-A/P-A

Control Program for TSCH blacklisting	SC: DLA+CLA G-A	SC: DLA+CLA BE-A/P-A	SC: DLA+CLA BE-A/P-A
Control program for STRALE (Standard-compliant & mobility-aware PHY rate & A-MPDU length adaptation)	TBD	TBD	TBD
GENERAL INTELLIGENCE FRAMEWORK			
Node-RED tool	SC/OC: Apache 2.0	SC/OC: Apache 2.0	SC/OC: Apache 2.0
Extensions of Node-RED with new services	IBM-CLA	IBM-CLA	IBM-CLA
SPECIFIC INTELLIGENCE COMPONENTS			
Data Collection Component			
Control Programs that specify the data that needs to be collected and the storage of the data and according metadata	SC: MIT [TUB] SC: DLA+CLA [IMINDS, CNIT] SC: LGPL or DLA+CLA [TCD] G-A	SC: MIT [TUB] SC: DLA+CLA [IMINDS, CNIT] SC: LGPL or DLA+CLA [TCD] BE-A/P-A	SC: MIT [TUB] SC: DLA+CLA [IMINDS, CNIT] SC: LGPL or DLA+CLA [TCD] BE-A/P-A
Data Aggregation Algorithms			
Algorithms with code developed from WiSHFUL partners	SC: MIT [TUB] SC: DLA+CLA [IMINDS, CNIT] SC: LGPL or DLA+CLA [TCD] G-A	SC: MIT [TUB] SC: DLA+CLA [IMINDS, CNIT] SC: LGPL or DLA+CLA [TCD] BE-A/P-A	SC: MIT [TUB] SC: DLA+CLA [IMINDS, CNIT] SC: LGPL or DLA+CLA [TCD] BE-A/P-A
Algorithms from external libraries	Licensing terms of external libraries are adopted	Licensing terms of external libraries are adopted	Licensing terms of external libraries are adopted
Intelligence Composition Component			
Existing implementations of machine learning algorithms	Licensing terms of external libraries are adopted	Licensing terms of external libraries are adopted	Licensing terms of external libraries are adopted
Prediction models, developed by WiSHFUL partners	SC: MIT [TUB] SC: DLA+CLA [IMINDS, CNIT] SC: LGPL or DLA+CLA [TCD] G-A	SC: MIT [TUB] SC: DLA+CLA [IMINDS, CNIT] SC: LGPL or DLA+CLA [TCD] BE-A/P-A	SC: MIT [TUB] SC: DLA+CLA [IMINDS, CNIT] SC: LGPL or DLA+CLA [TCD] BE-A/P-A
Action Components			
Intelligent Control Programs that generate UPI commands for configuration of radio and network settings	SC: MIT [TUB] SC: DLA+CLA [IMINDS, CNIT] SC: LGPL or DLA+CLA [TCD] G-A	SC: MIT [TUB] SC: DLA+CLA [IMINDS, CNIT] SC: LGPL or DLA+CLA [TCD] BE-A/P-A	SC: MIT [TUB] SC: DLA+CLA [IMINDS, CNIT] SC: LGPL or DLA+CLA [TCD] BE-A/P-A
TESTBED EXTENSIONS			

FED4FIRE compliant testbed toolset			
jFed	SC: MIT G-A	SC: MIT BE-A / P-A	SC: MIT BE-A / P-A
Ansible	SC: GPL G-A	SC: GPL BE-A / P-A	SC: GPL BE-A / P-A
OMF6	SC: MIT G-A	SC: MIT BE-A / P-A	SC: MIT BE-A / P-A
OML	SC: MIT G-A	SC: MIT BE-A / P-A	SC: MIT BE-A / P-A
Portable testbed			
Emulab	SC: AGPLv3 G-A	SC: AGPLv3 BE-A / P-A	SC: AGPLv3 BE-A / P-A
OMF6	SC: MIT G-A	SC: MIT BE-A / P-A	SC: MIT BE-A / P-A
OML	SC: MIT G-A	SC: MIT BE-A / P-A	SC: MIT BE-A / P-A
Backbone Network Control Program	SC: MIT G-A	SC: MIT BE-A	SC: MIT BE-A
Wireless mesh solution for routing in backbone network	OC: DLA G-A	OC: DLA BE-A / P-A	OC: DLA BE-A / P-A
Generic data visualization tool			
PHP back end	SC: MIT G-A	SC: MIT BE-A / P-A	SC: MIT BE-A / P-A
HTML5 front end	SC: MIT G-A	SC: MIT BE-A / P-A	SC: MIT BE-A / P-A

Table 21: Licensing models used in WiSHFUL

4 Data Management Plan

As the WiSHFUL project has decided to make its results public, this section includes the first version of the Data Management Plan describing the Project's policy and practices regarding the provision of Open Access to each of the public deliverables, presentations, scientific publications and research data sets that it will produce.

The ***main obligations*** of participating in the Open Research Data Pilot are:

1. To make it possible for External Parties to *access, mine, exploit, reproduce and disseminate* - free of charge for any user - the following:
   - the data, including associated metadata, needed to validate the results presented in scientific publications, as soon as possible
   - other data, including associated metadata, as specified and within the deadlines laid down in the data management plan; and
2. To provide information about *tools* and *instruments* at the disposal of the beneficiaries and necessary for validating the results.

In order to meet these obligations, this document follows the Guidelines on Data Management in Horizon 2020 [12] and describes the ***data management life cycle*** for all data sets to be collected, processed or generated by the project according to the following schema:

- Data set reference and name (identifier)
- Data set description
- Associated standards and metadata
- How the data will be shared
- How the data will be archived and preserved

The following ***categories of outputs*** require Open Access to be provided - free of charge - by the WiSHFUL partners, including organisations selected through its Open Calls, to related data sets in order to fulfil the H2020 requirements of making it possible for External Parties to access, mine, exploit, reproduce and disseminate the results contained therein:

- (public) Deliverables
- Conference/Workshop presentations (which may, or may not, be accompanied by papers, see below)
- Conference/Workshop papers and articles for specialist magazines
- Research Data and metadata

Information will also be given about *tools* and *instruments* at the disposal of the beneficiaries and necessary for validating the results.

The document is organised as follows: Sections 4.1, 4.2 and 4.3 deal with the data management life cycles of deliverables, presentations and scientific publications, respectively. Section 4.4 describes where metadata associated with research results can be found for enabling External Parties to *access*, *mine*, *exploit*, *reproduce* and *disseminate* the results, free of charge. This document will continually evolve during the lifespan of the Project as new public deliverables, presentations, scientific publications and research data sets are released, and will take into account any changes in consortium practices and external factors.

4.1 Open access to public deliverables

4.1.1 Data Sharing

Open Access to its public deliverables is achieved in WiSHFUL by depositing the data into an online repository. The public deliverables are stored in one or more of the following locations:

- The WiSHFUL Website, after approval by the project (if the document is subsequently updated, the original version will be replaced by the latest version), at this URL: http://www.wishful-project.eu/deliverables
- The Cordis web site, will host all public deliverables as submitted to the European Commission. The WiSHFUL page on Cordis is: http://cordis.europa.eu/project/rcn/194304_en.html

Public deliverable license is currently under discussion in the consortium, the two options considered are:

- Attribution-ShareAlike 4.0 International (http://creativecommons.org/licenses/by-sa/4.0/), that allows for sharing content from the deliverables under the same license if the original content is cited and allows for commercial re-usage;
- Attribution-NonCommercial-ShareAlike 4.0 International (https://creativecommons.org/licenses/by-nc-sa/4.0/), that allows for sharing content from the deliverables under the same license if the original content is cited and does not allow for commercial re-usage.

4.1.2 Archiving and Preservation

Open Access to project public deliverables will be maintained for at least 3 years following the project completion, through the Website.

4.1.3 Deliverable Registration Template

An up to date table with archived public deliverables can be found at http://www.wishful-project.eu/deliverables

4.2 Open access to conference/workshop presentations

4.2.1 Data Sharing

Open Access to conference/workshop presentations is achieved in WiSHFUL by depositing the data into an online research data repository. The presentations are stored in the following location:

- The WiSHFUL Website at this URL: http://www.wishful-project.eu/events

Public presentation license is currently under discussion in the consortium, the two options considered are:

- Attribution-ShareAlike 4.0 International (http://creativecommons.org/licenses/by-sa/4.0/), that allows for sharing content from the deliverables under the same license if the original content is cited and allows for commercial re-usage;
- Attribution-NonCommercial-ShareAlike 4.0 International (https://creativecommons.org/licenses/by-nc-sa/4.0/), that allows for sharing content from the deliverables under the same license if the original content is cited and does not allow for commercial re-usage.

4.2.2 Archiving and Preservation

Open Access to project public presentations will be maintained for at least 3 years following the project completion, through the Website.

4.2.3 Presentation Registration Template

An up to date table with archived public presentations can be found at http://www.wishful-project.eu/events.

4.3 Open access to scientific publications

During the project's three-year lifespan, WiSHFUL will produce a number of technical documents and publications in conference/workshop proceedings and articles published in magazines. There are two main routes to providing Open Access to these publications, namely, ´gold´ or ´green´.

The ***'gold' Open Access*** approach to peer-reviewed scientific research articles means that an article is immediately provided in Open Access mode by the scientific publisher. The associated costs are shifted away from readers. The most common business model is based on one-off payments by authors. These costs, often referred to as Article Processing Charges (APCs), are usually borne by the researcher's university or research institute or the agency funding the research. In other cases, the costs of Open Access publishing are covered by subsidies or other funding models.

The ***'green' Open Access*** approach to peer-reviewed scientific research articles means that the author, or a representative, self-archives (deposits) the published article or the final peer-reviewed manuscript in an *online repository* before, at the same time as, or after publication. Some publishers request that Open Access be granted only after an *embargo period* has elapsed. This *embargo period* is to allow the scientific publisher to recoup their investment by selling subscriptions and charging pay-per-download/view fees during an exclusivity period. In the case of WiSHFUL, this *online repository* may be the project Website, or (for example) a pre-print archive.

4.3.1 Publication Reference Identity

The DOI (Digital Object Identifier) uniquely identifies a document. This will be allocated by the publisher, in the case of ´gold´ Open Access, or by OpenAIRE in the case that the document is archived in ZENODO.

4.3.2 Data Sharing

Open Access to its publications will be achieved in WiSHFUL by depositing the data into an online research data repository. The publications will be stored in one or more of the following locations: data:

- An institutional research data repository
- The ZENODO repository, operated by the European Commission funded OpenAIRE project
- The WiSHFUL Website, at the url: http://www.wishful-project.eu/publications
- Other data repositories (searchable by, for example: http://www.re3data.org)

The ZENODO repository (http://www.zenodo.org/) is the one mainly recommended by the EC's OpenAIRE initiative in order to unite all the research results arising from EC funded projects.

ZENODO is an easy-to-use and innovative service that enables researchers, EU projects and research institutions to share and showcase multidisciplinary research results (data and publications) that are not part of existing institutional or subject-based repositories. Namely, ZENODO enables users to:

- easily share the long tail of small data sets in a wide variety of formats, including text, spreadsheets, audio, video, and images across all fields of science
- display and curate research results, get credited by making the research results citable, and integrate them into existing reporting lines to funding agencies like the European Commission
- easily access and reuse shared research results
- define the different licenses and access levels that will be provided

Furthermore, ZENODO assigns a Digital Object Identifier (DOI) to all publicly available uploads, in order to make content easily and uniquely citable and this repository also makes use of the OAIPMH protocol (Open Archives Initiative Protocol for Metadata Harvesting) to facilitate the content search through the use of defined metadata. This metadata follows the schema defined in INVENIO[1] (a free software suite enabling to run an own digital library or document repository on the web) and is exported in several standard formats such as MARCXML, Dublin Core and DataCite Metadata Schema according to OpenAIRE Guidelines.

In addition, considering ZENODO as the repository, the short- and long-term storage of the research data will be secured since they are stored safely in same cloud infrastructure as research data from CERN's Large Hadron Collider. Furthermore, it uses digital preservation strategies to storage multiple online replicas and to back up the files (Data files and metadata are backed up on a nightly basis).

Therefore, this repository fulfils the main requirements imposed by the EC for data sharing, archiving and preservation of the data generated in H2020 projects.

4.3.3 Archiving and Preservation

Open Access to project public presentations will be maintained for at least 3 years following the project completion, through the Website.

4.3.4 Publication Registration Template

An up to date table with archived public presentations can be found at http://www.wishful-project.eu/publications.

4.4 Open access to research data

The scientific and technical results of the WiSHFUL project are expected to be of high interest for the scientific community. Through the duration of the project, once the relevant protections (e.g. IPR) are secured, the WiSHFUL partners are disseminating (subject to their legitimate interests) the obtained results and knowledge to the relevant scientific communities through contributions in journals and international conferences in the field of wireless communications, including cognitive radio.

Apart from the Open Access to public deliverables, presentations and scientific publications explained in the previous sections, the Open Research Data Pilot also applies to two ***types of data***[2]:

[1] http://invenio-software.org/

[2] EC document: "Guidelines on Open Access to Scientific Publications and Research Data in Horizon 2020" - version 1.0 - 11 December, 2013

- The data, including associated metadata, needed to validate the results presented in scientific publications (underlying data);
- Other data, including associated metadata, as specified and within the deadlines laid down in a data management plan, to be developed by the project. In other words, beneficiaries will be able to choose which data, additionally to the data underlying publications, they make available in Open Access mode.

According to this requirement, the underlying data related to the scientific publications will be made publicly available (see section 4.4.4). This will allow that other researchers can make use of that information to validate the results, thus being a starting point for their investigations, as expected by the EC through its Open Access policy.

These data will include a description of the procedures followed to obtain those results (e.g., software used for simulations, experimental setups, equipment used, etc.) as well as data generated following those procedures (experimental measurements results, spreadsheets, images, etc.).

In addition, other type of data generated during the project could be the specifications of the WiSHFUL software platform and the services it supports, the datasheets and performances of the technological developments of the project, the showcase results with the KPIs (Key Performance Indicators) used to evaluate the system performances, among others.

Since a huge amount of data is generated in a European project as WiSHFUL, we will make a selection of relevant information, disregarding data not being relevant for the validation of the relevant published results. Moreover, we will analyse on a case-by-case basis all data generated during the project before making them open in order to be aligned with the exploitation and protection policy. As a result, the publication of research data will be mainly followed by those partners involved in the scientific development of the project (i.e., academic and research partners), while those partners focused on the "development" of the technology will limit this publication of information due to strategic/organizational reasons (commercial exploitation).

A more detailed description of the information expected to be generated in WiSHFUL and whether and how it will be exploited or made publicly available is provided in Section 4.4.4.

4.4.1 Metadata

Metadata refers to "data about data", i.e., it is the information that describes the data that is being published with sufficient context or instructions to be intelligible for other users. Metadata must allow a proper organization, search and access to the generated information and can be used to identify and locate the data via a web browser or web-based catalogue.

Two types of metadata will be considered within the frame of the WiSHFUL project: that corresponding to the project public deliverables, presentations and scientific publications, which has already been described in sections 4.1, 4.2 and 4.3, respectively, and that corresponding to the published research data (this section).

In the context of data management, metadata will form a subset of data documentation that will explain the purpose, origin, description, time reference, creator, access conditions and terms of use of a data collection.

The metadata that would best describe the data depends on the nature of the data. For research data generated in WiSHFUL, it is difficult to establish a global criteria for all data, since the nature of the initially considered data sets will be different, so that the metadata will be based on a generalised metadata schema as the one used in ZENODO, which includes elements such as:

- ***Title***: free text
- ***Creator***: Last name, first name

- ***Date***
- ***Contributor***: It can provide information referred to the EU funding and to the WiSHFUL project itself; mainly, the terms "European Union (EU)" and "Horizon 2020", as well as the name of the action, acronym and the grant number
- ***Subject***: Choice of keywords and classifications
- ***Description***: Text explaining the content of the data set and other contextual information needed for the correct interpretation of the data
- ***Format***: Details of the file format
- ***Resource Type***: data set, image, audio, etc.
- ***Identifier***: DOI
- ***Access rights***: closed access, embargoed access, restricted access, open access.

Additionally, a readme.txt file could be used as an established way of accounting for all the files and folders comprising the project and explaining how all the files that make up the data set relate to each other, what format they are in or whether particular files are intended to replace other files, etc.

4.4.2 Data Sharing

Open Access to its research data will be achieved in WiSHFUL by depositing the data into an online research data repository. The research data will be stored in one or more of the following locations: data:

- An institutional research data repository
- The ZENODO repository, operated by the European Commission funded OpenAIRE project
- The WiSHFUL Website: will at least provide link(s) to the repository/repositories where research data is stored
- Other data repositories (searchable by, for example: http://www.re3data.org)

4.4.3 Archiving and Preservation

Open Access to research data will be maintained for at least 3 years following the project completion.

4.4.4 Research Data Set Registration Template

This section provides an explanation of the different types of data sets to be produced in WiSHFUL, which has been identified at this stage of the project. As the nature and extent of these data sets can evolve during the project, more detailed descriptions will be provided in future versions of the DMP.

The descriptions of the different data sets, including their reference, file format, the level of access, and metadata and repository to be used (considerations described in Section 4.4.1 and 4.4.2), are provided in table with the following entries for documenting the archived data sets:

- Data set reference
- Data set name
- Data set description, including
  - description of the data
  - origin of data (if collected)
  - nature and scale and to whom it could be useful, and whether it underpins a scientific publication

- information on the existence (or not) of similar data and the possibilities for integration and reuse.

- File format
- Standards and metadata: describe the contents of the data files and the context in which they have been established
- Repository Location:

As off today, no data sets are provided yet, but in the context of the Intelligence framework, large data sets are being produced for offline wireless network optimization.

5 Conclusions

Related to **guidelines for IPR management** of the software developed within the WiSHFUL Project, we have proposed a legal framework for access and use of the WiSHFUL software by External Parties during and beyond the Project.

First, a survey of the software assets and their owners is presented. A distinction is made between assets that already existed at the start of the project, referred to as 'basic software platforms', and assets that arise from the WiSHFUL Project. For the existing software platforms, we started from the initial legal ownership and Background, as stipulated by the Grant Agreement (GA) and refined in the Consortium Agreement (CA). For the software assets that arise from the WiSHFUL project, ownership of the Project Results is assigned as a function of the work generated in WiSHFUL.

Next, a summary is given of the initial legal framework for agreements between consortium partners about access rights dealing with the question how to offer Background and Results related to the WiSHFUL Project from one WiSHFUL partner to other WiSHFUL partner(s) in order to develop a practical joint solution ("access rights for implementation"). This initial legal framework also defines how access to Background and Results related to the WiSHFUL Project is given to External Parties ("access rights for exploitation of the Results"), that use the WiSHFUL software either as a Third Party in a funded Open Call, or a non-funded External Party.

Finally, this document proposes a more detailed legal framework for access and use of the WiSHFUL software by WiSHFUL partners, Third Parties or other External Parties, during or beyond the project, taking into account the specifics of the developed software during the course of the WiSHFUL project. Three scenarios for accessing the WiSHFUL software have been identified: (1) Access by Third Parties funded through WiSHFUL Open Calls during the WiSHFUL project; (2) Access by non-funded External Parties during and beyond the WiSHFUL Project; and (3) Access by WiSHFUL partners beyond the WiSHFUL Project. We further distinguish three access rules defining the conditions for access in terms of charges for using software, amount of technical support, and quotas for using experimental facilities: (1) guaranteed access, (2) best effort access and (3) premium access. In view of sustainable use of the WiSHFUL software, we support different licensing models for access to object code and source code for the three different use scenarios. The early definition of transparent conditions for the access rights to object and source code of WiSHFUL software will facilitate the access and use by Third Parties and other External Parties during and after the project. Most software tools developed in the WiSHFUL project appear to be free for non-commercial academic research and can be used under well-defined licensing models. No major issues with access rights have been identified so far.

Related to the **Data Management Plan**, various outputs will be offered for Open Access by means of the following procedures:

- ***Deliverables*** designated as "Public" will be placed on the project's public website in the menu category "Documents->Deliverables" (https://wishful-project.eu/deliverables/)
- Conference/Workshop ***presentations*** will be placed on the project's public website in the menu category "Events" http://www.wishful-project.eu/events)
- The copyright of conference/workshop ***papers*** ("scientific publications" category on the WiSHFUL public website) may be owned by the scientific publisher. In this case, the scientific publisher may allow the associated costs for accessing the paper to be shifted away from readers through a one-off payment (Article Processing Charge - APC) by the author. This ('gold' Open Access) scheme will be considered, knowing that these costs can be recovered from the agency funding the research. Alternatively, the author, or a representative, may self-archive (deposit) the published article or the final peer-reviewed manuscript in an *online*

repository before, at the same time as, or after publication. This ('green' Open Access) scheme will be used by some partners in the project. Some publishers request that Open Access be granted only after an *embargo period* has elapsed. This *embargo period* is to allow the scientific publisher to recoup their investment by selling subscriptions and charging pay-per-download/view fees during an exclusivity period

- ***Articles*** accepted for publication in magazines will be handled in the same way as scientific publications.
- ***Research data*** (such as underlying data and associated metadata in scientific publications) will be archived and preserved on ZENODO. The intention is that other experimenters can then repeat the tests and reproduce the results

6 References

[1] I. Tinnirello, G. Bianchi, P. Gallo, D. Garlisi, F. Giuliano, and F. Gringoli, "Method of changing the operation of wireless network nodes," Sept. 29 2015. US Patent 9,148,800.

[2] Python-based ecosystem of open-source software for mathematics, science, and engineering, http://www.scipy.org

[3] Python Data Analysis Library, http://pandas.pydata.org

[4] scikit tools for data mining and data analysis, http://scikit-learn.github.io/stable

[5] Data Mining Software in Java, http://www.cs.waikato.ac.nz/ml/weka/

[6] International Business Machines, Inc. (IBM), Individual Contributor License Agreement, http://nodered.org/cla/node-red-cla-individual.pdf

[7] Apache License, Version 2.0, January 2004, http://www.apache.org/licenses/

[8] MIT License, https://opensource.org/licenses/MIT

[9] GNU General Public License, http://www.gnu.org/copyleft/gpl.html

[10] GNU Lesser General Public License, http://www.gnu.org/licenses/lgpl.html

[11] Open Source Initiative, Berkeley Software Distribution, http://opensource.org/licenses/bsd-license.php

[12] European Commission, Guidelines on Data Management in Horizon 2020, 15 February 2016. Available at: http://ec.europa.eu/research/participants/data/ref/h2020/grants_manual/hi/oa_pilot/h2020-hi-oa-data-mgt_en.pdf

Appendix 1: Dual licensing agreement for commercial use and online academic use (DLA)

Thank you for your interest in the [Software – add name of software or software tool]

If you continue to log in, you will be granted access to the [Software] for academic use. For licensing opportunities for commercial use of the [Software], please contact [Owners'] IPR Office.

Upon log in you confirm that you agree with the licensing terms below.

For us it is essential that:

- **You will only use the [Software] for academic purposes**
- **If you publish an article that contains results you obtained by using the [Software], you will cite the following publication [add full reference to publication].**
- **You will not engage in a funded project using the [Software] without consulting us**

The [Software] is owned by **[Owner],** [legal status of organisation], having its administrative offices in [add full address] and is provided in a time limited license to registered users for non-commercial research evaluation purposes only. The following is a legally binding agreement between yourself as licensee of the [Software] and [Owner] as the legal owner and licensor of the Software.

By downloading, installing and/or using the [Software], you agree to the following terms and conditions.

If you do not agree to the license terms as described hereunder, you should not download, run, use or install the [Software].

DEFINITIONS

In this Agreement the following expressions (either in singular or plural and with first letter capitalised) have the meaning as follows:

"**Agreement**" shall mean this software license agreement.

"**Software**" shall mean the [Software] as used, downloaded or installed by the user.

"**Purpose**" shall mean the use of the Software for non-commercial research purposes only.

"**Licensor**" shall mean [Owner]

Grant of rights, Exclusion of rights, Proprietary Rights

License. Subject to Licensee's continuous compliance with the terms and conditions of this Agreement, Licensor grants to Licensee a non-exclusive, non-assignable license to use the Software for the Purpose in a manner consistent with the terms of this Agreement and applicable documentation.

- **Citation.** Licensee will cite the following publication "**[add full reference to publication]"** in every publication that contains results that were obtained using the Software.

Restrictions. No other rights, even implied, are granted herein, other than those mentioned in clause 2.1 under this Agreement. The Licensee is specifically **not** allowed to:

a) modify, adapt, translate, reverse engineer or create derivative works based on the Software or knowingly permit any third party to engage in the foregoing.

b) sublicense, distribute, disclose, market, rent, lease, or transfer the Software.

c) copy or to allow others to make or obtain copies of the Software.

d) provide services to third parties with the Software.

e) Use the Software in funded projects

Ownership. As between Licensor and Licensee, the Licensor controls all right, title and interest in the Software.

Trade marks. Licensee shall not adopt, use or register any trademark, trade name or other marketing name of Licensor, or any confusingly similar trademark, trade name or marketing name except if permitted by Licensor in writing. Licensee shall not remove any trademark notice from the Software.

TERM AND TERMINATION

Term. This Agreement shall come into force on the date of you first download, use or installing the Software and shall continue in full force and effect for a term of one year ("**License Term**"). Upon the expiration of the License Term, some or all of the Software may cease to operate without prior notice. Upon expiration or termination of the License Term, Licensee may not use the Software.

Return. Upon termination, Licensee shall return the Software to Licensor if delivered to Licensee via a physical data carrier or format; Licensee shall immediately cease any use of the Software and shall remove any copy of (fragments of) the Software from any system under its control.

LIMITED WARRANTIES – LIABILITIES

Licensor has no obligation to support or maintain the Software and grants Licensee the right to use the Software "as is". Licensee assumes total responsibility and risk for Licensee's use of the Software.

Licensor does not make, and expressly disclaims, any express or implied warranties, representations or endorsements of any kind whatsoever, including, without limitation, the implied warranties of merchantability or fitness for a particular purpose, and the warranties of title or non-infringement.

In no event shall Licensor be liable for (**a**) any damages (including, without limitation, damages for loss of profits, business interruption, loss of programs or information, and the like) arising out of the use of or inability to use the Software, even if Licensor has been advised of the possibility of such damages, (**b**) any claim attributable to errors, omissions, or other inaccuracies in the Software, or (**c**) any claim by any third party. By way of example only, Licensor does not warrant that the Software will be error-free. As used in this section, "Licensor" includes its employees, directors, officers, agents, representatives, subcontractors, service providers and suppliers.

GENERAL

Assignment: Licensee may not assign or transfer this Agreement as a whole, or any of its rights or obligations under it, without first obtaining the written consent of Licensor. That consent may not be unreasonably withheld or delayed.

Illegal/unenforceable provisions: If the whole or any part of any provision of this Agreement is void or unenforceable in any jurisdiction, the other provisions of this Agreement, and the rest of the void or unenforceable provision, will continue in force in that jurisdiction, and the validity and enforceability of that provision in any other jurisdiction will not be affected.

Waiver of rights: If Licensor fails to enforce, or delays in enforcing, an obligation of the Licensee, or fails to exercise, or delays in exercising, a right under this Agreement, that failure or delay will not affect its right to enforce that obligation or constitute a waiver of that right. Any waiver of any

provision of this Agreement will not, unless expressly stated to the contrary, constitute a waiver of that provision on a future occasion.

No agency: Nothing in this Agreement creates, implies or evidences any partnership or joint venture between the parties, or the relationship between them of principal and agent. Neither party has any authority to make any representation or commitment, or to incur any liability, on behalf of the other.

Entire agreement: This Agreement constitutes the entire agreement between the parties with respect to the subject matter and supersedes all prior arrangements, understandings, representation and communications, oral or written with respect to the subject matter.

Formalities: Each party will take any action and execute any document reasonably required by the other party to give effect to any of its rights under this Agreement.

Amendments: No variation or amendment of this Agreement will be effective unless it is made in writing and signed by each party's representative.

Governing law: This Agreement is governed by, and is to be construed in accordance with [add country] Law. The courts of Ghent will have exclusive jurisdiction to deal with any dispute that has arisen or may arise out of or in connection with this Agreement, except that either party may bring proceedings for an injunction in any jurisdiction.

Appendix 2: Contributor License Agreement (CLA)

Project [Software – add name of software or software tool]

The [Software] Project is a joint effort by contributors from across many countries. In order to have a clear IP situation that benefits the contributors as well as the Beneficiaries, Contributor agrees to submit his/her Contribution under the terms of this Contributor License Agreement.

You agree with this Contributor License Agreement by confirming it with the accept button, by pushing your code to github, by sending the code to one of [Software] core developers, or by distributing your code to the [Software] Project in any other way. By doing so you enter into a legally binding agreement with:

- [Owner 1], [Full address], registered under [VAT number];
- [Owner 2], [Full address], registered under [VAT number];
- ...

In this document referred to as the "Beneficiaries".

YOU AGREE AS FOLLOWS:

1. Definitions

When used in this Contributor License Agreement the following words and or expressions shall have the meaning as stated hereunder:

- "Contributor" means any individual and/or legal entity that voluntarily submits (a) Contribution(s) to the [Software] Project.
- "Contribution" means any work, including any modification and/or addition to the existing work that is submitted for introduction in, or documentation of, the [Software] Project, where such work originates from a Contributor. A Contribution may be submitted in any form of electronic, verbal and/or written communication or documentation, including without limitation, communication on electronic mailing lists, source code control systems and issue tracking systems that are managed by or on behalf of the [Software] Project for the purpose of discussion and improving the results of the [Software] Project.

.

Article 2: Granted Rights - Representations

For the benefit of the Beneficiaries, the Contributor hereby:

(i) irrevocably assigns, transfers and conveys to the Beneficiaries all right, title and interest in and to the Contributions. Such assignment includes copyrights and all other intellectual property rights, together with all causes of actions accrued in favour for infringement thereof, recognized by any jurisdiction ("Proprietary Rights"). The assignment is made for the entire duration of their protection and worldwide. Without limitation of the foregoing, the Beneficiaries shall be entitled to determine in their sole discretion whether or not to use the Contribution and to use, sell, distribute, license, re-produce, re-use, modify, update, edit or otherwise make available the Contribution(s) as they see fit, in any manner currently known or in the future discovered and for any and all purposes.

(ii) grants (to the extent that under applicable mandatory law, Proprietary Rights cannot be assigned, transferred or conveyed) to the Beneficiaries and to the recipients of the software incorporating the Contribution(s) an irrevocable, worldwide, non-exclusive, fully paid-up and royalty-free copyright license for the entire term of copyright protection to reproduce, modify, prepare derivative works of, (publicly) display, perform, sublicense and distribute the Contribution; and

(iii) grants to the Beneficiaries and to recipients of software distributed by the Beneficiaries a worldwide, non-exclusive, fully paid-up, royalty-free, irrevocable (except as stated in this Agreement) Patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Contribution, where such license applies only to the Patent claims licensable by Contributor that are necessarily infringed by the Contributor's Contribution(s) alone or by combination of such Contribution(s) with other work of the Beneficiaries. Contributor furthermore agrees to immediately notify the Beneficiaries of any patents that Contributor knows or comes to know are likely infringed by the Contribution(s) and/or are not licensable by the Contributor. If any entity institutes patent litigation against the Contributor or any other entity (including a cross-claim or counterclaim in a lawsuit) alleging that Contributor's Contribution(s) or the [Software] Project work to which the Contributor has contributed constitutes direct or contributory patent infringement, then any Patent licenses granted under this Agreement for that Contribution or [Software] Project work shall immediately terminate as of the date such litigation is filed.

Upon the assignment of the Proprietary Rights and the grant of the license as set forth in this article 2, the Beneficiaries hereby grant to the Contributor a non-exclusive, worldwide, fully-paid up, royalty-free license to make, use reproduce, distribute, modify and prepare derivative works based on the Contributions of Contributor.

Should the Beneficiaries decide not to include the Contribution in the [Software] Project within three months after submission, the Contributor is entitled to request the Beneficiaries to cancel the contribution in order to allow the Contributor to re-acquire all right, title and interest in the cancelled Contribution, without effect on the non-cancelled Contributions. The Beneficiaries will not unreasonable withhold their agreement to cancel.

Contributor hereby represents and warrants that:

i. Contributor is legally entitled to assign the Proprietary Rights and to grant the above license. If Contributor's employer(s) has rights to intellectual property that Contributor creates that includes Contributor's Contributions, Contributor represents and warrants that Contributor has received permission to make Contributions on behalf of its employer, that its employer has waived such rights for such Contributions to the Beneficiaries;

ii. each Contribution is the original creation of the Contributor. Contributor represents that each submission of a contribution includes complete details of any third-party license or other restrictions of which you are aware and which are associated with any part of the Contribution(s);

iii. (to the best of your knowledge, no claim or dispute has been threatened or filed in connection with the ownership, use or distribution of the Contribution(s); and

iv. the execution of this Contributors License Agreement does not constitute a breach under any other agreement or duty to which Contributor and/or its employer is a party, does not require the consent, approval or waiver from or notice to any third party and does not violate any law or regulation.

Contributor shall immediately inform the Beneficiaries of any facts and/or circumstances of which Contributor becomes aware that would make the representations and warranties inaccurate or untrue in any respect. Contributor further agrees that Contributor shall at no time hereafter dispute, contest or aid or assist a third party in disputing and/or contesting, either directly or indirectly, the right, title and interest in any and all Contributions of the Beneficiaries as detailed in this Contributor License Agreement.

In case that under applicable mandatory law the Contributor retains the moral rights or other inalienable rights to the Contributions, the Contributor agrees not to exercise such rights without the prior written permission of the Beneficiaries.

In order to ensure that the Beneficiaries will be able to acquire, use and protect the Proprietary Rights as detailed in this article 2, Contributor will (i) sign any documents to assist the Beneficiaries in the documentation, perfection and enforcement of these rights, and (ii) provide the Beneficiaries with support and reasonable access to information for applying, securing, protecting, perfecting and enforcing these rights.

The rights granted under this Agreement are granted to both Beneficiaries individually, who can individually exercise these rights.

Article 3: Warranties

EXCEPT FOR THE EXPRESS WARRANTIES DETAILED IN ARTICLE 2, THE CONTRIBUTIONS ARE PROVIDED "AS IS" AND NEITHER CONTRIBUTOR NOR THE BENEFICIARY MAKE ANY WARRANTIES OF ANY KIND TO THE OTHER PARTY, EITHER EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION OF ANY WARRANTY OF MERCHANTAIBILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Article 4: Miscellaneous

This Contributor License Agreement shall enter into force upon execution of this document by Contributor. This Contributor License Agreement may be terminated by a party if the other party commits a breach of this Contributor License Agreement provided that if the breach is capable of remedy termination shall not occur if the breach shall not have been remedied within 90 days of such other party having been given notice in writing specifying the breach and requiring it to be remedied. The termination of this Contributor License Agreement shall however remain in full force and effect with respect to any Contribution submitted prior to the termination date of the Agreement.

This Contributor License Agreement contains the entire agreement between the parties and supersedes all prior or contemporaneous agreements or understanding, whether written or oral, relating to its subject matter. If any provision of this Contributor License Agreement shall be deemed invalid or unenforceable, the validity and enforceability of the remaining provisions of this Agreement shall not be affected and such provision shall be deemed modified only to the extent necessary to make such provision consistent with applicable law.

The Contributor License Agreement is governed by the laws of [Country], without reference to its conflict of law principles.