D1.2 “PERIODIC REPORT (DRAFT)”

Version 1.3

Document Information

Contract Number	780681
Project Website	https://legato-project.eu/
Contractual Deadline	31st of May of 2019
Dissemination Level	Confidential
Nature	Report
Authors	Osman Unsal, Sergi Madonar, Jens Hagemeyer, Marcelo Pasin, Miquel Pericas, Mustafa Abduljabbar, Micha vor dem Berge, Rose Gregorio and Zeba Shamshad
Contributors	All partners
Reviewers	Osman Unsal and Sergi Madonar

The LEGaTO project has received funding from the European Union’s Horizon 2020 research and innovation programme under the Grant Agreement No 780681

Change Log

Version	Description of Change
V 1.0	Initial draft for internal review
V1.1	Inputs from the partners
V1.2	Addition of the estimated financial information
V1.3	Revision

Index

1. Explanation of the work carried out by the beneficiaries and Overview of the progress 4
   1.1 Objectives, LEGaTO System Stack and Technical Progress 4
      Project Objectives 4
      LEGaTO Project Components 4
      M0-M9 Technical Progress 6
      M9-M18 Technical Progress and Future Timeline 6
   1.2 Explanation of the work carried per Work Package 7
      1.2.1 Work Package 1 – Project Management and Coordination 7
      1.2.2 Work Package 2 – Hardware Platform 9
      1.2.3 Work Package 3 – Tool-chain Back End 14
      1.2.4 Work Package 4 – Tool-chain Front End 22
      1.2.5 Work Package 5 – Application development and optimization 30
      1.2.6 Work Package 6 – Project Dissemination and Exploitation 35
   1.3 Impact 36
2. Update of the plan for exploitation and dissemination of result 37
3. Follow-up of recommendations and comments from previous review(s) 38
4. Deviations from Annex 1 and Annex 2 38
5. Reporting 39
   5.1 Personnel costs 39
   5.2 Other Direct Costs 40
   5.3 Person Months 41

Annex I: Addressed Recommendations 42

1. Explanation of the work carried out by the beneficiaries and Overview of the progress

1.1 Objectives, LEGaTO System Stack and Technical Progress

Project Objectives

The primary ambition of the LEGaTO project is to start with a Made-in-Europe mature software stack, and optimize this stack to support energy-efficient computing on a commercial cutting-edge European-developed CPU–GPU–FPGA heterogeneous hardware substrate and FPGA-based Dataflow Engines (DFE), which will lead to an order of magnitude increase in energy efficiency.

The final outcome of the LEGaTO Project will be a complete software toolset that includes a programming environment and an execution framework for energy-efficient execution of concurrent applications running on state-of-the-art het-heterogeneous hardware substrate.

The primary ambition of an order of magnitude energy savings described above will drive the project towards the final expected outcome. The project also aims to increase programmer productivity, increase fault-tolerance and ensure a trusted and secure compute substrate; achieving all these goals in an energy-efficient manner.

LEGaTO Project Components

To set the stage, we include a brief reminder of the LEGaTO stack in Figure 1.1. In addition to the description of the particular stack level (hardware, middleware, runtime, applications), Work Package identifiers were also included in the figure.

The LEGaTO project will apply the energy-efficient software toolset for heterogeneous hardware to five applications. The first application will be healthcare. The project will not only demonstrate a decrease in energy consumption in the healthcare sector; it will also show that the toolset will increase healthcare application resilience and security; both of which are critical requirements in this area. In three further applications, the project will demonstrate ease of programming and energy savings possible through the use of the LEGaTO project software–hardware framework for secure IoT gateways, smart homes, and smart city applications. The fifth application will be based on machine learning (ML), where the project will demonstrate how to improve energy efficiency by employing accelerators and tuning the accuracy of computations at runtime. In addition, the machine learning use case will be used to further optimize the energy efficiency in the two other use cases, as well as within the runtime.

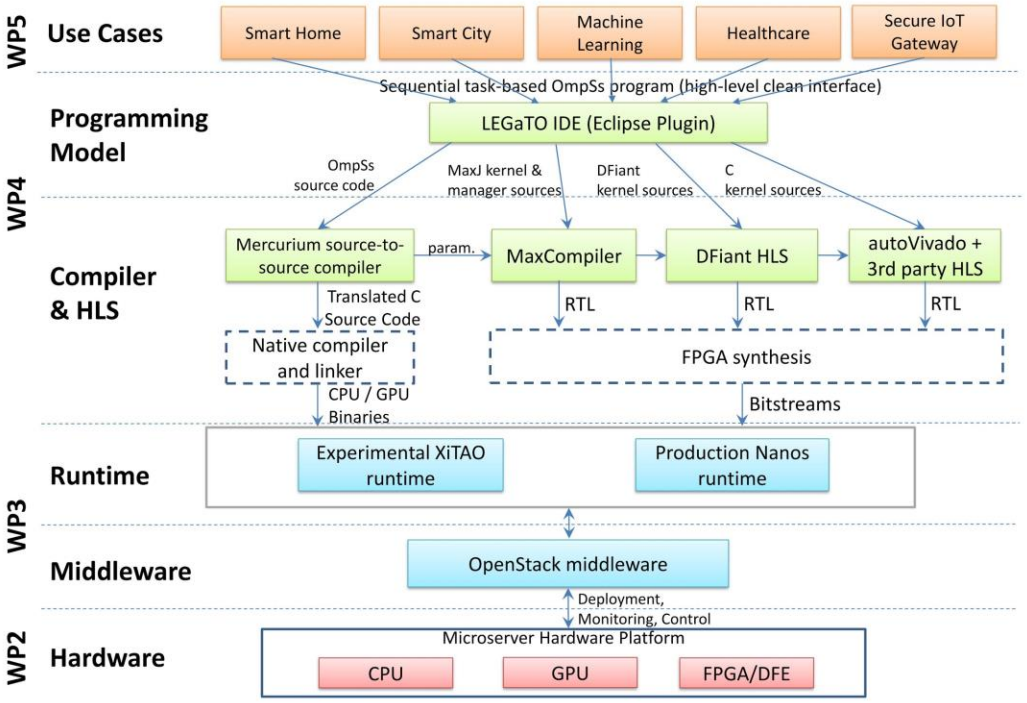


Figure 1. The LEGaTO stack at a glance

On the software side, LEGaTO will have a task-based toolset to start with. This task-based toolset, has three pillars: at the higher-level the tool-chain is driven by the task-based OmpSs Programming Model, which is an extension of OpenMP intended to influence future generations of the standard. This is coupled to the Mercurium compiler which can handle CPUs, and GPUs, and is being extended to handle FPGAs. The third element of the pillar is the Nanos runtime, which schedules tasks to compute elements in data-flow fashion. One ambition of the LEGaTO project is to enhance this tool-chain with energy-efficiency transformations and elevate the enhanced tool-chain to a high maturity level through an IDE release by the end of the project. In the project we will develop and integrate OmpSs@FPGA (providing support for FPGAs) and OmpSs@Cluster (providing a shared memory look and feel over and application spread over multiple nodes).we A major research direction is to make tasks moldable, which enables a more efficient resource management and sharing. Such generalized tasks are currently part of the task-based runtime titled XiTAO. By the end of the project, the promising research ideas from XiTAO will be integrated to the common runtime. A major research direction is to make tasks moldable, which enables a more efficient resource management and sharing. Such generalized tasks are currently part of the task-based runtime titled XiTAO. By the end of the project, the promising research ideas from XiTAO will be integrated to the common runtime.

Among the LEGaTO hardware substrates, FPGAs provide both a promise and challenge. On one side, FPGAs are very power ecient, and can substantially accelerate such applications as Deep Neural Network inference (which will be deployed as part of the ML use-case by Data Intelligence). On the other side, FPGAs have been notoriously difficult to program. This will be addressed in LEGaTO through the DFiant high level synthesis (HLS) language available to develop key kernels for the FPGA fabric; the project will integrate these kernels into the common LEGaTO runtime by the end of the project.

This hardware platform features GPU and FPGA accelerators that are already energy efficient. Even so, by the end of the project, we are planning to further expand the energy

efficiency of this heterogeneous platform by endowing it with a hardware interconnection fabric. Moreover, the microclusters in this heterogeneous platform will be dynamically composed as requested by the LEGaTO runtime. OpenStack will be extended to support these dynamic compositions of microclusters, incorporate the specific needs of the different heterogeneous microservers as well as the communication infrastructure between them. Using OpenStack as a middleware layer allows accessing the hardware and deploying the applications in a standard uniform way, supporting cloud computing use cases.

The second heterogeneous platform is based on Dataflow Engines (DFEs). DFEs are also based on FPGAs, but their architecture is highly optimised for throughput-oriented computations of large data sets, making them ideally suited for HPC and big-data workloads. OmpSs will provide support to ooad large, static sub-graphs to DFE accelerators. These will be then compiled into DFE kernels with the MaxCompiler toolchain.

M0-M9 Technical Progress

In the period, all technical WPs made good overall progress, we divide the period into a) M0 to M9 (interim review timeline) and b) M9 to M18 (second review timeline) and highlight the following progress (additional progresses not highlighted here are described in each WP).

For M0 to M9, WP5 has produced specifications for all application use cases, including requirements, bottlenecks, metrics and optimisation plans. WP4 progressed in the definition of the front-end tool box, including programming model (annotations for fault-tolerance, energy requirements, etc.), execution framework and languages. WP3 defined the back-end runtime system with capabilities for FPGA support, execution management and reconfiguration, and hardware backend interfaces. WP2 produced a detailed hardware and firmware specification for LEGaTO, including definitions for the IoT gateway, network infrastructure, reconfigurable hardware support and the microserver testbed.

M9-M18 Technical Progress and Future Timeline

For M9 to M18, in WP5 the smart city kernel is ported to OmpSs and tested on ARM64 platforms. For the Machine Learning use case, a first implementation of an OmpSs taskified artificial neural network model has been done for CPU and this implementation has also been carried over to OmpSs@FPGA. For WP4, we incorporated OpenMP and OmpSs support into Eclipse IDE. We also developed a secure checkpointing mechanism, and conducted analysis of interplay of energy versus security in ARM trustzone. On WP3, we developed backend drivers for nanos-6 runtime to support GPU and FPGA platforms and implemented a first integration between OmpSs@Cluster and OmpSs@FPGA (currently under test). In WP3, we also pushed the first public release of the XiTAO runtime, which features an elastic resource mapper and a scheduler that anticipates hardware heterogeneity. An implementation of VGG-16 Deep Neural Network targeting XiTAO is also provided. In WP2, the dynamic node composition has been integrated into Redfish and OpenStack, providing the required abstraction towards the run-time in WP3. Furthermore, in parallel to advancement in the edge server development, we are contributing to a new computer-on-module form factor for edge computing named COM HPC, which is standardized by the PICMG consortium.

In addition to the project timeline in the DoW, and based on Interim Review feedback, we established a more detailed timeline with additional internal checkpoints. According to this timeline, we plan to release OmpSs@FPGA, OmpSs@Cluster, XiTao, Maxcompiler and

Dfiant at Month 20, integrate OmpSs with XiTao, Maxcompiler and Dfiant at Month 24, apply runtime reconfigurability and OmpSs@Cluster on ARM/X86 on Month 30, and finally integrate LEGaTO HW, programming model and use cases by Month 36.

1.2 Explanation of the work carried per Work Package

1.2.1 Work Package 1 – Project Management and Coordination

Task 1.1: Administrative and financial management

The project management structure and procedures are described in detail in section 3 Implementation of the Annex 1 (part A). All procedures described there have been implemented at an early stage of the project, proved to work as designed and to be appropriate for a project of this size and ambition.

The General Assembly has met five times:

- The 14th of December of 2017 in Barcelona, Spain (kick off meeting).
- The 19th of February of 2018 in Champery, Switzerland.
- The 4th of July of 2018 in Gothenburg, Sweden.
- The 26th of November of 2018 in Brussels, Belgium (Interim Review).
- The 9th of April of 2019 in Tel Aviv, Israel.

A consortium agreement (CA) was prepared and agreed with the partners. Among others, the CA governs the responsibilities of the partners, the liability, the management structure, rules for decision-making and conflict solving, financial provisions and payments as well as the IPR.

Budget and distribution of funds

The first action was the distribution of the advance payment according to the Consortium Agreement. The following table shows the distribution of the pre-financing and foreseen payments among partners.

Partner	Direct Personnel Costs	Other Direct Costs	Indirect Costs	Total	Prefinancing 75%	Guarantee fund 5%	Rest 20%
BSC	725.420,00 €	145.642,50 €	217.766,00 €	**1.088.828,50 €**	816.621,37 €	54.441,43 €	217.765,70 €
UNIBI	457.650,00 €	65.700,00 €	130.838,00 €	**654.188,00 €**	490.641,00 €	32.709,40 €	130.837,60 €
UNINE	336.000,00 €	25.700,00 €	90.425,00 €	**452.125,00 €**	339.093,75 €	22.606,25 €	90.425,00 €
CHALMERS	489.483,00 €	33.850,00 €	130.833,00 €	**654.166,00 €**	490.624,50 €	32.708,30 €	130.833,20 €
DIS	114.000,00 €	28.200,00 €	35.550,00 €	**177.750,00 €**	133.312,50 €	8.887,50 €	35.550,00 €
TUD	319.200,00 €	25.700,00 €	86.225,00 €	**431.125,00 €**	323.343,75 €	21.556,25 €	86.225,00 €
CHR	558.600,00 €	89.200,00 €	161.950,00 €	**809.750,00 €**	607.312,50 €	40.487,50 €	161.950,00 €
HZI	137.400,00 €	29.200,00 €	41.650,00 €	**208.250,00 €**	156.187,50 €	10.412,50 €	41.650,00 €
TECHNION	324.000,00 €	38.200,00 €	90.550,00 €	**452.750,00 €**	339.562,50 €	22.637,50 €	90.550,00 €
MAXELER	440.000,00 €	28.200,00 €	117.050,00 €	**585.250,00 €**	438.937,50 €	29.262,50 €	117.050,00 €
	3.901.753,00 €	**509.592,50 €**	**1.102.837,00 €**	**5.514.182,50 €**	**4.135.636,87 €**	**275.709,13 €**	**1.102.836,50 €**
						5.514.182,50 €	

In addition to the periodic reporting obligations towards the European Commission, a regular reporting scheme on 6-month basis has been established and implemented. This scheme enables the project management to follow the project closely in terms of resources and allows comparing the spent resources with the progress.

Task 1.2: Technical coordination

The technical coordination is based on regular meetings between the coordinator and the WP leaders as well as on parallel meetings with each WP participants.

The LEGaTO project has an Industrial Advisory Board (IAB) with the objective of ensuring the quality of the results of the research activities of the project. The current members of the IAB are the following:

Sector HW/ SW	Member	Institution	Country	Justification	Industry	Std. Body	Policy maker
HW	Michaela Blott	Xilinx Inc.	Ireland	Machine Learning, Data centers and FPGAs.	X		
HW	Stephan Diestelhorst	ARM	UK	Computer architecture, energy monitoring and modelling	X		
HW & SW	Marius Feldman	Cloud & Heat	Germany	Energy efficient cloud computing, green computing	X		
HW & SW	Ayal Zaks	Intel	Israel	Compiler Optimizations, Parallel architectures	X	X	
SW	Mariano Lamarca	Barcelona City Council	Spain	Networking, smart cities			X
HW & SW	Prof. Dr. Ingmar Steinhart	v. Bodelschwingh Foundation Bethel	Germany	Health care, social service	X		
HW & SW	*TBC*	Continental AG	Germany	Low power computing, automobiles	X		

Task 1.3: Internal communication, Quality and Risk management

Internal communication within the project

The principle to follow in the monitoring process is that the lowest adequate level has to decide, taking into account the guidance of the Project Management Office (PMO) at BSC. The Coordinator is the responsible of making progress reports and ensuring timely preparation of deliverables. Based on this and due to the number of WP in this project the monitoring of work progress in each work package has been delegate to WP leaders. They have established their tools and communication strategy with the partners involved in each WP. During this period, the coordinator has also hold several technical calls separately with each WPs to be sure that activities of different WPs are aligned.

The consortium holds a teleconference meeting every month. The scientific work is monitored through these meetings as well as the preparation of periodic reports. Decisions taken in the meetings are described in the minutes available in the SVN for the whole consortium.

Several mailing lists have been set per work package and the general ones in order to ensure the information reaches to all partners.

Internal communication with the EC

The coordinator and the Project Officer have had a fluent communication and any change and/or delay of milestones and deliverables has been informed. All communications are shared with the consortium.

Quality and risk management

The preparation of deliverables is responsibility of WP leaders. A set of templates has been prepared and they are available in the project SVN.

The quality assessment of the deliverables/milestones includes a peer review by other partners. A draft of the deliverable/milestone should be made available to the WP leader 15 days before the deadline. This draft will be circulated to the partners, who will have 15 days to grant approval or suggest modifications. The templates have a control sheet to grant approval.

Based on the above-mentioned periodic meetings, foreseen risks are monitored monthly and if needed mitigation action will be implemented.

1.2.2 Work Package 2 – Hardware Platform

The hardware platform of LEGaTO, defined, developed and maintained in Work Package 2, provides the lowest layer and the base for the LEGaTO flow. The LEGaTO hardware supports the full range of heterogeneous microserver technology from CPUs to FPGAs for cloud as well as edge use cases. The flexibility of the hardware platform provides an optimal match for a wide range of use cases, in addition, it offers a flexible high-speed, low-latency communication infrastructure.

The work package is split into five tasks, covering the specification of the hardware platform (T2.1) as well as the testbed (T2.2), followed by the hardware development tasks centered around hardware design of cloud and edge components (T2.3), as well as providing the required integration in the LEGaTO toolchain (T2.4 – T2.5). The hardware specification has been finalized in Deliverable D2.1, which was included in the D2.1 SD1 (Chapter 6).

Apart from the state which has been reported in SD1, and was finalized in by month 9 of the project, significant progress to all of the tasks has made, except for T2.1, which has been finished. The testbed (T2.2) has been upgraded with new microservers based on Jetson AGX Xavier and new Intel FPGAs. A significant amount of work was dedicated to edge server development (T2.3). New microserver form factors have been defined and are currently standardized in the COM HPC workgroup of the PICMG consortium, ensuring wide industry coverage. The microserver baseboard is under development considering OCP along with other form factors. The firmware and middleware (T2.4) have been extended by support for dynamic node composition, which is represented and abstracted by Redfish and OpenStack towards the run-time in WP3. The node composition mechanisms also include dynamic reconfiguration of the communication infrastructure (T2.5). New drivers and deployment mechanisms are currently under development to support this feature. In addition, benchmarks and analysis of the PCIe-based host-2-host communication have been conducted to analyze the performance for different microservers. The detailed progress of the different tasks is reported in the following sections.

Task 2.1: Hardware and firmware specification

Within task 2.1, the hardware platform of the LEGaTO project has been defined. The LEGaTO hardware platform supports both, cloud as well as edge use cases, and enables deployment of the LEGaTO stack for various use cases. Two cloud platforms are used in LEGaTO, the RECS microserver platform as well as Maxeler's DFE. In addition, an edge platform is developed within the project. The platforms support the full range of heterogeneous microserver technology from CPUs to FPGAs, which can be seamlessly combined to provide the optimal platform for a given use case or application. The integrated high-speed, low-latency communication infrastructure provides another powerful optimization option. The dynamic node composition feature, which is controlled by the management functionality in combination with the Redfish and OpenStack middleware, allows adapting the communication topology at run-time, dynamically adapting to changing application requirements. Below, an overview of the different microservers used in LEGaTO is given. For each microserver from CPU to FPGA, a low-power, as well as a high-performance variant, is supported. This provides the possibility to apply a concept like big-little, currently employed in many ARM-based SoCs for the mobile market, on the system level across different architectures.

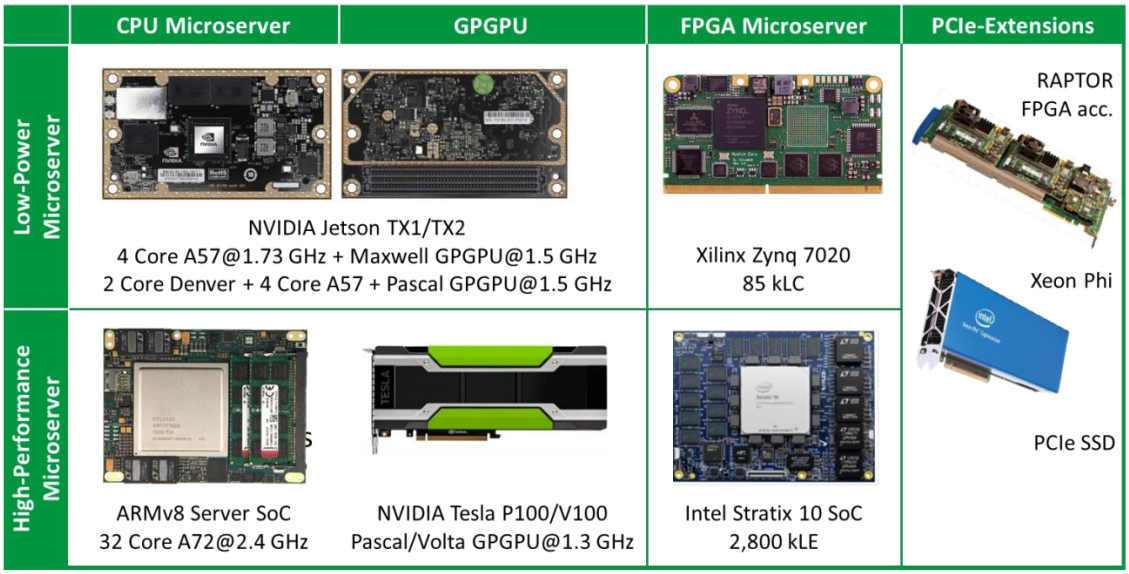


Figure 2: RECS Microserver Overview

Task 2.2: Testbed setup and maintenance

The project provides central testbeds for test and deployment of the LEGaTO toolchain, as well as application development and benchmarking. The main testbed which supports all microservers introduced above is currently hosted at Poznan supercomputing center (in collaboration with the M2DC EU project). Furthermore, local development systems are used at each partner for easy test and verification, as well as specialized hardware features like FPGA undervolting.

Figure 1: RECS Server chassis

In addition to the main project testbed, a Maxeler MPC-X testbed (Figure 3) is hosted at Jülich Supercomputer center, which is accessible to all project partners. It provides a high-end FPGA accelerator solution as well as the MAX-J toolflow, which is currently integrated in the LEGaTO toolchain. Maxeler provides tool support, maintenance and training for this system.

Figure 2: Maxeler MPC-X system

Task 2.3: Development of optimised hardware components

This task focuses on optimization of hardware components for cloud as well as edge use cases. While the cloud hardware is focused on the data center, the edge server architecture supports applications with local (pre)processing requirement, for data-reduction, direct user interaction, faster response time or safety/security/dependability reasons. The architecture developed within LEGaTO features a modular hardware design and support for all three major compute architectures, i.e. CPU, GPU, and FPGA. It is based on the existing microservers and form factors, which are also used in the modular, cloud-based architecture. In addition to the existing form factors, also new developments like Jetson AGX Xavier from NVIDIA or new PICMG standards are supported as well. The architecture supports a distributed setup including a reasonably fast, low-latency communication link between the different units. At the current state, the development of the edge server platform is ongoing, an architectural diagram is provided in Figure 4.

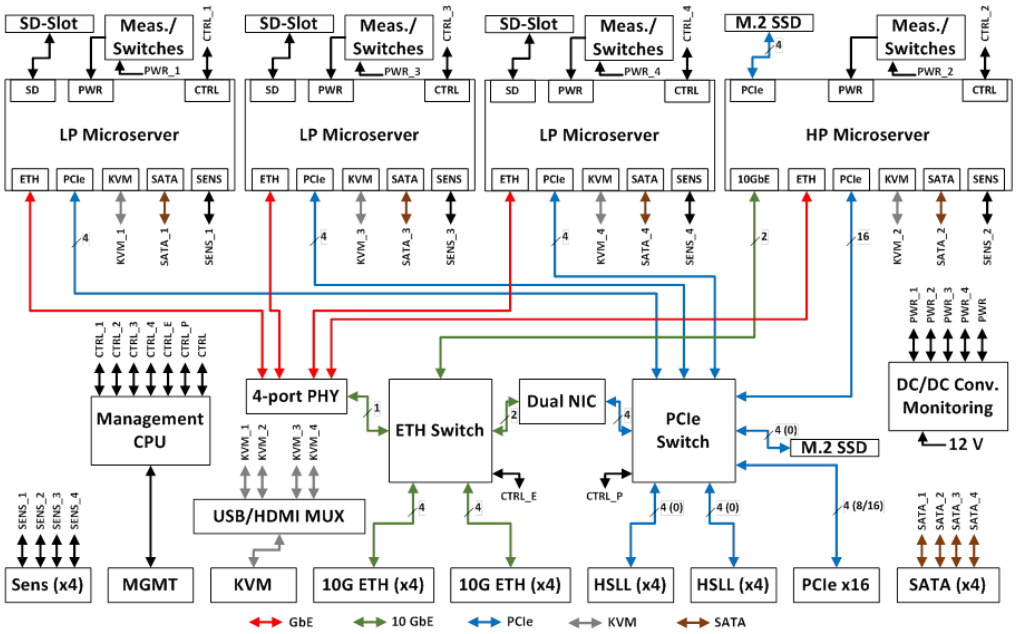


Figure 3: Proposed edge server architecture

In addition to the edge server development, a TCO analysis of the cloud platform has been performed. As the overall cloud platform is a modular system, parts which are not required for a particular use-case can be removed or redesigned for lower cost. A direct comparison between the non-optimized and a TCO optimized version of the chassis shows CAPEX savings of about 15 % while OPEX savings are about 7,5 % due to reducing energy usage overhead of the removed infrastructure.

Task 2.4: Firmware and low-level software

This task is about the development of firmware and low-level software for the heterogeneous hardware. The focus lies on the definition of a Redfish API and its implementation within the RECS_Master management software. This enables static node composition and dynamic reconfiguration of the communication infrastructure at runtime. Node composition allows the bundling of hardware resources, such as servers and accelerators, exclusively together by interconnecting them through the high-speed, low-latency communication infrastructure and giving them access to available PCIe functions. This communication structure can then be reconfigured at runtime.

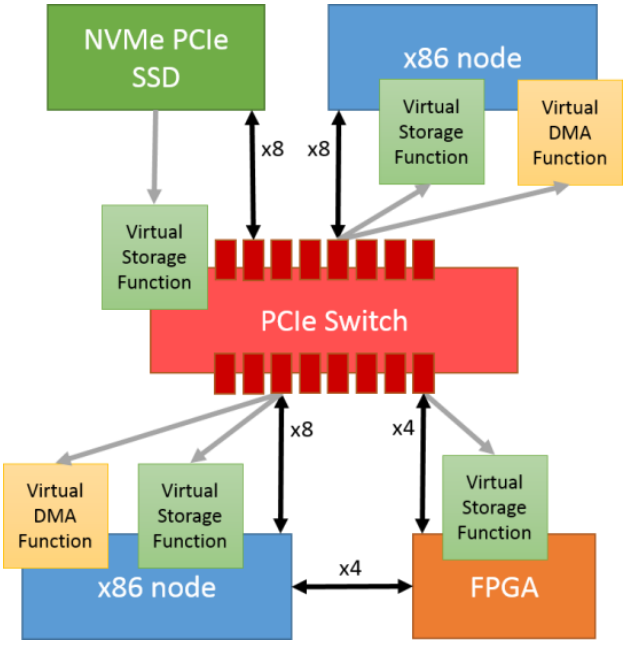


Figure 4: Node composition example

The redefinition of a basic Redfish API to support enhanced node composition (see Figure 5 for an example) and dynamic reconfiguration is completed. It provides access to the aforementioned composition features and can be triggered by upper middleware and application layers to control the heterogeneous hardware infrastructure. The implementation of this API within the RECS_Master management software is also finished.

We will now concentrate on continuing the improvement of the maturity and robustness level of the firmware and low-level software considering possible further extensions to meet additional requirements from partners.

Task 2.5: Infrastructure for FPGA-based computing

Within Task 2.5, a communication infrastructure with a focus on FPGA-based computing is developed. For this, the RECS platform supports two basic features for high-speed, low-latency communication. On the one hand, a communication infrastructure based direct, raw high-speed serial links is supported, which provides highest speed and lowest latency, e.g., for FPGA-based communication using optimized low-level protocols. On the other hand, PCIe communication across different compute nodes is supported. In contrast to typical implementations, in which PCIe is used to connect peripherals to a host, the RECS communication infrastructure can be used for host-2-host communication, providing a flexible communication link. From a software point of view, the connection can be used like a network interface, providing a socket-based connection. Figure 6 shows a benchmark result using iperf tests, showing a useable bandwidth of 56 Gbit/s. The latency of the connection is about four times lower compared to a regular 10 Gbit/s Ethernet connection.

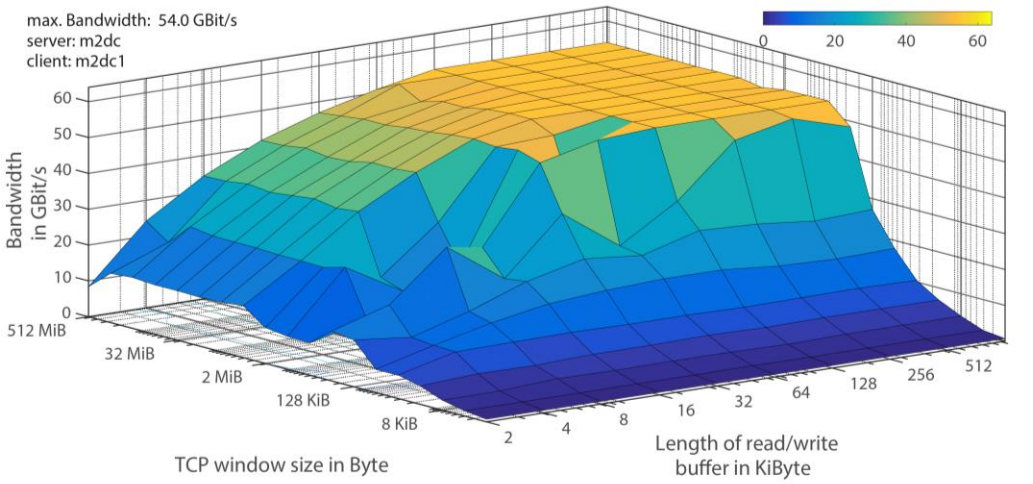


Figure 5: PCIe Host-2-Host communication performance

1.2.3 Work Package 3 – Tool-chain Back End

Work Package 3 focuses on the runtime layer to support the compilation and the execution of secure, resilient and energy efficient applications. This section summarises the technical progress done in Work Package 3, during the first 18 months of LEGaTO.

When designing a global architecture encompassing the entire LEGaTO tool chain to support applications, we chose to split them into front-end and back-end components. The back-end components are developed in WP3, while the front-end ones are developed in Work Package 4. Some components may have parts both in the back-end and in the front-end of the tool chain, and for the sake of clarity and coherence, we decided to present them either in Work Package 3 or in Work Package 4.

The development of building blocks in Work Package 3 was split into six tasks: definition/design, middleware and backend drivers, software and hardware topologies, energy-efficient runtime, performance and debug tools, runtime support for fault-tolerance and security. Task T3.1 (definition/design) has finished, while the progress of all the other tasks are ongoing.

Work Package 3 ended its first reporting period during the preparation of Deliverable D3.2, First release of the task-based runtime, due month 20.

Task 3.1: Definition / Design

Task 3.1 has defined the design of the back-end runtime system for LEGaTO, in which multiple heterogeneous devices and multiple resource management libraries are targeted using task offloading according to Figure 7. The task defines the LEGaTO runtime and proposes scheduling technologies targeting low energy consumption. This includes a proposal for the structure of the wrapper function to be generated from the Mercurium compiler for integration of Xilinx FPGAs, through the Vivado HLS tool. For cluster, an Execution Workflow to be implemented in OmpSs-2 (Nanos-6) has been proposed, including the support for memory transfers and task offloading. For XiTAO, we have defined novel software topologies to be used for locality-aware scheduling, which can be used to drive the reconfiguration at the hardware level. In addition, XiTAO integrates novel scheduling techniques for low-energy computing. In addition, T3.1 has defined tools for improving the

performance and correctness of LEGaTO applications. These tools provide online monitoring of application execution and provide access to debugging facilities across the heterogeneous hardware stack. We also define functionality to support fault tolerance. While such functionality has been researched extensively in the context of CPUs, LEGaTO is implementing checkpointing functionality for FPGAs and GPUs, as well as task replication.

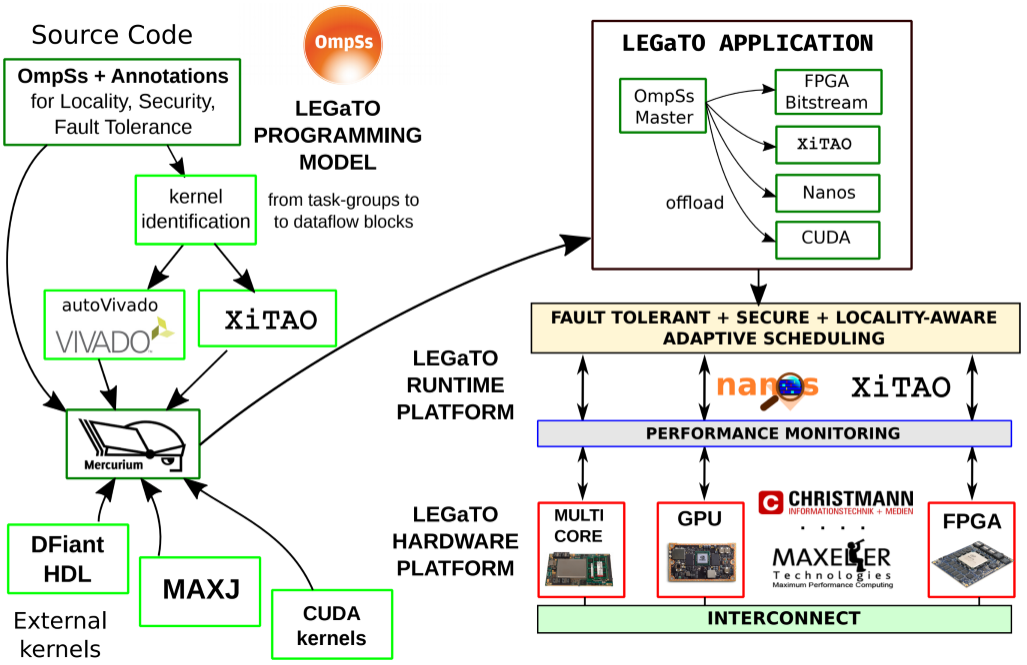


Figure 7: Overview of the back-end design

Task 3.2: Middleware and backend drivers

The development within task 3.2 provides the middleware layer of the LEGaTO software stack. It enables upper layers to interact with the complex heterogeneous hardware, either directly or through OpenStack, to ease its usage and thus broaden the targeted commu nities and markets. It also provides inventory information to be used in task 3.3.

The task consists of two sub-tasks. The first one is the development of an OpenStack extension to configure and reconfigure the underlying hardware to meet the requirements of tasks by the runtime system. It interfaces with the Redfish API, developed in T2.4. Still, after the expert's feedback of the first review, it was decided that the development of this OpenStack extension will be carried out with a low priority in favour of extending firmware and Web GUI of the RECS|Box Server itself, which is the second task. The goal is to allow easy and full-featured (re-)configuration of the underlying hardware, even without OpenStack. The so far developed OpenStack extension will still be supported but not updated throughout the project.

The second sub-task is about developing back-end drivers for nanos-6 to manage resources within the operation systems and react to reconfiguration requests appropriately.

OpenStack middleware

The first steps of the OpenStack sub-task were the analysation of OpenStack, in order to identify and understand the components needed for the LEGaTO middleware layer. In addition to the main components such as Nova, Neutron, Keystone and Horizon, Ironic will be used for bare-metal provisioning, Cyborg for accelerator Management and Valence for

the dynamic management of pooled resources. The latter will enable OpenStack to control the heterogeneous hardware by composing nodes consisting of multiple resources (Servers, Accelerators).

After the analysation phase, a test environment with the OpenStack Queens release was set up and later reinstalled with the later Rocky release with Ansible. In that course, some additional Ansible and Bash scripts were developed and configured to enable an automatic roll out of most OpenStack parts.

Furthermore, the development of an adapted version of OpenStack Valence has been carried out in collaboration with the M2DC project. At this time, it supports basic static node composition by calling the Redfish API, developed in T2.4. As stated above, the development of this plugin is paused at the moment in favour of a direct support within the Server's firmware.

Backend drivers

We have developed the runtime system support for GPUs and FPGAs. On GPUs, CUDA and OpenCL libraries from vendors provide the necessary low level services for Nanos-6. On the Xilinx FPGAs, we have provided the xdma and xtasks libraries that work on top of the vendor driver in Linux.

We currently interoperate with the Xilinx and Alpha-Data drivers on the various Xilinx integrated and discrete FPGAs that we support (Zynq 7000, Zynq U+, and Virtex-7).

Task 3.3: Software and hardware topologies

Task T3.3 includes the definition of the interface between the hardware middleware based on OpenStack and the runtimes being developed in LEGaTO. This task has two parts:

First is the development of a layer to obtain hardware information from OpenStack and run programs on the RECS®|Box 4.0 platform, which is a heterogeneous hyperscale server developed by CHR and UNIBI.

Second is the development of runtime-level APIs for locality aware execution by defining a notion of locality among the tasks to be executed by the task-parallel runtimes in LEGaTO.

Rigorous Planning of the First Task

In terms of the first task, no actual development has yet started, but we have had multiple discussions to plan how to approach it. We observed that the OpenStack infrastructure for hardware inventory management is based on the hwloc library[1]. To our advantage, this will allow the runtimes to obtain infrastructure information from such a popular library. Additionally, this is the preferred method for both Nanos and XiTAO. For the first toolchain deliverable (M20), we do not plan to support dynamic reconfiguration of the RECS hardware. Instead, virtual images will be booted on the RECS nodes, and the local per-node configuration will be read by the applications during runtime initialization. Locality-aware policies implemented in the runtimes will then be used to schedule the applications in a locality efficient manner.

[1] According to https://specs.openstack.org/openstack/fuel-specs/specs/9.0/support-numa-cpu-pinning.html

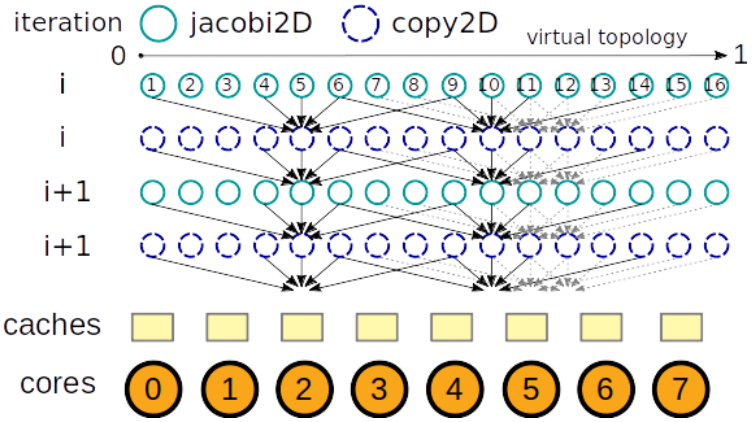


Figure 8: Virtual topology mapping of Jacobi2D and Copy2D kernels

Virtual Hardware Topologies in XiTAO

The second task has two components, one for XiTAO and the other for Nanos. Once a set of tasks is executed in Nanos or XiTAO, locality-aware policies implemented in the runtime will guide the execution of tasks to reduce data communication and hence reduce energy consumption. We begin by discussing the facility implemented in XiTAO. At the task level, XiTAO implements a concept called "virtual topologies" which is then –at runtime- converted into actual core mappings to enforce locality aware scheduling. XiTAO's virtual topologies consist of regular N-dimensional cartesian topologies. Figure 8 shows the virtual mappings of jacobi2D and copy2D kernels. Each task is optionally given an address in the virtual topology. By measuring the virtual distance between two XiTAO tasks, the runtime obtains approximate information on the communication relationship between the two tasks. If the two tasks have the same address, this is understood by the runtime as meaning high amount of data reuse between the two tasks. As a consequence, the XiTAO runtime attempts to schedule the two tasks on the same set of cores. This optimistically results in data reuse via the caches of the cores. In the current state of XiTAO, we have implemented one-dimensional virtual topologies and verified the positive outcome on performance using a heat diffusion simulation benchmark.

Task 3.4: Energy efficient runtime

The task T3.4 is the main task focusing on the development of the LEGaTO runtimes. It has three main components: the development of XiTAO, the development of OmpSs-2 (OmpSs@cluster) and the development of OmpSs@FPGA.

XiTAO

XiTAO is the main experimental runtime developed within LEGaTO. An initial version of XiTAO existed before the start of the LEGaTO project. This initial version had several limitations: it targeted only homogeneous architectures, it required programmers to manually tune hardware resources, and its only goal was performance. During the first half of the LEGaTO project, all these limitations have been overcome

XiTAO now features an online scheduler for static and dynamic heterogeneity

To uplift the scope of the XiTAO runtime to target diverse low-energy scheduling, an extensible online scheduler is under development. The current progress can be summarized in the following 2 miletones.

1. Introducing the Performance Trace Table (PTT) to Enhance Heterogenous Scheduling

Heterogeneity is a major target of the LEGaTO project. During the early months of the LEGaTO roject, a master thesis conducted by two students at Chalmers (Agnes Rohlin and Henrik Fahlgren) took the homogeneous version of XiTAO[2] and ported it to the HiKey 960 board, a big.LITTLE platform with four A73 (big) cores and four A53 (LITTLE) cores. In addition, an online performance monitor was developed. To enhance heterogeneity, the scheduler encapsulates a Performance Trace Table (PTT). For each type of tasks, PTT records the execution time of the last task's execution time and constructs a weighted average over the previous executions similar to the figure below. The table is used to schedule high-priority tasks at runtime[3]. It determines the appropriate core and number of resources that such tasks should use in order to minimize the execution time of the application. This scheduler targets both static heterogeneity (e.g., big.LITTLE hardware) as well as dynamic heterogeneity such as runtime interference or system-level DVFS activity.

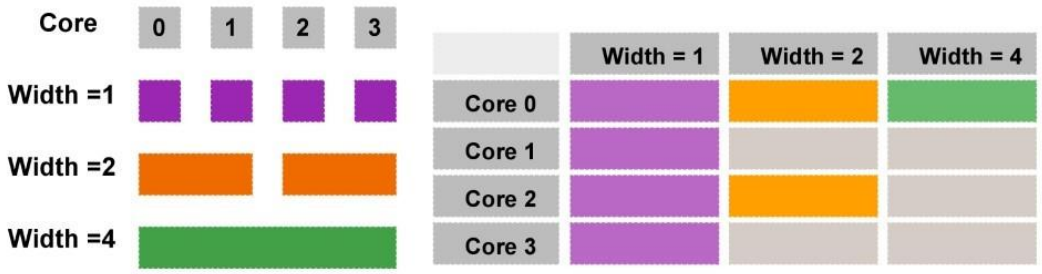


Figure 9: Overview of XiTAO's Performance Trace Table (PTT)

2. Energy-Aware Scheduling on the TX2 Platform. Documenting and Publicly Releasing XiTAO

After milestone 1, a new PhD student and a postdoc were hired for the project to proceed with the development of the runtime. Their recent progress has resulted in several outcomes. First, the XiTAO code has been considerably cleaned up and the project has been released as open source[4]. Next, the code has been ported to the TX2 platform, which is one of the platforms adopted by the LEGaTO project and included in the RECS hardware. Finally, a detailed power model is being developed for the TX2 platform. We intend to use this power model to develop an energy-aware scheduler that, in addition to performance, targets energy minimization and minimization of the Energy-Delay Product (EDP). This will be hopefully completed within the next few weeks. Finally, to provide new test cases for the XiTAO runtime, we have developed an implementation of VGG-16 targeting XiTAO. This implementation has been shown to be competitive compared to the original OpenMP implementation when executing on CPU cores. As part of the current work, we are extending this implementation to use also the CUDA cores. This will allow us to develop the first heterogeneous implementation of XiTAO targeting not only heterogeneous

[2] By homogenous we mean that the resources have uniform performance, and resources are therefore uniformly assigned to tasks.
[3] Such tasks are marked with high-priority if they fall on a critical path, which is determined based on a graph algorithm that will described in a research paper.
[4] Available on https://github.com/mpericas/xitao.

microarchitectures (such as big.LITTLE) but also heterogeneous hardware architectures (such as CPU+GPU).

OmpSs@Cluster

In the context of LEGaTO, we modify OmpSs-2@Cluster in order to allow integration for supporting accelerators, i.e. OmpSs-2@CUDA, OmpSs-2@FPGAs. Accordingly, transparent to the user, we introduce conceptual model of the steps required to execute a task which, in Nanos6 terminology is called the Execution Workflow. In order to execute a task, once a scheduling decision has been made, Nanos6 creates an Execution Workflow for that task, which is a dependency graph that consists of a number of Execution Steps forming a small dependency graph. These Execution Steps are:

1) a number of *Allocation and Pinning* steps, which allocate the memory necessary to store the data accesses of the task on the device (SMP, NVIDIA, GPU, FPGA) which will execute it.

2) a number of **Data Copy** steps which program the copies needed for the input accesses of the task on the device.

3) an **Execution** step which actually executes the task on the device,

4) a number of **Data Release** steps which release the dependencies of the data accesses of the task and potentially the memory that stores them

5) a **Notification** step which notifies the runtime about the completion of the task.

Each Execution Step starts only after all the predecessor steps have completed, e.g. the Execution step will start only after all Data Copy steps are completed, etc. These abstract Steps are implemented by each device that Nanos6 supports. That being the case, each device can choose whether a Step will be executed synchronously or asynchronously, depending on the device capabilities.

Whenever possible, we try to implement these steps in an asynchronous fashion, so that we can overlap communication with computation as much as possible. Cluster support in Nanos6 has been designed from scratch so as to use the Execute Workflow design. Porting CUDA and FPGA support is also underway and initial integration being between Cluster and these accelerators has been achieved. Moreover, we have extended Nanos6 support for cluster-related Extrae trace generation tool events. This integration is crucial in understanding bottlenecks related specifically with cluster. Tracking these bottlenecks is essential for optimizing Cluster for performance and scalability.

Task 3.5: Performance and debug tools

We have improved the FPGA instrumentation infrastructure to obtain accurate information about the internals of the execution of tasks and data transfers to/from FPGAs.

We have started the development of a debugging tool to trace the correctness of the task dependencies. This tool is based on the Pin instrumentation tool from Intel. Pin captures the memory accesses performed by application tasks, and our tool compares them with the task-declared input and output memory regions. Then the tool can detect mismatches among this information, and the actual memory accesses. It also detects race conditions among tasks.

Task 3.6: Runtime support for fault-tolerance and security

FPGA Undervolting

We have aimed to evaluate the energy-resilience trade-off of the LEGaTO system through aggressive supply voltage underscaling below the default level. It has been shown on multiple devices like CPUs, GPUs, FPGAs, DRAMs, and SRAMs, that aggressive undervolting can deliver significant energy saving; however, as a downside, in below the minimum safe voltage level, i.e., Vmin, timing faults can appear due to the circuit delay path increase. To alleviate this issue, fault mitigation techniques like frequency underscaling, ECC, CRC, among others, can play a crucial role to improve the resilience of the whole system.

Progress

Our primary focus is on the FPGAs since their utilization in the modern data center is rapidly growing; however, their energy efficiency is still a key concern especially when compared against equivalent ASICs. We have already performed a comprehensive work on the evaluation of aggressive voltage underscaling on Xilinx platforms. Our study included voltage guardband analysis, fault characterization, fault mitigations using ECC, and Neural Network (NN) evaluation.

More specifically, the overall voltage behavior observed for a FPGA-based NN is illustrated in Fig. 1. As seen, by voltage underscaling below the default level, i.e., Vnom, there is a Guardband region. In this region, there is energy efficiency improvement without compromising the NN accuracy or performance since no faults appear. By further undervolting below the guardband and due to the circuit delay path increase, faults start to appear at V1stfault. However, until a minimum safe voltage level, i.e., Vmin, relatively low fault rate does not impact the application due to the natural resilience of NN applications and thus, there is no accuracy loss, i.e., Silent region. By further voltage underscaling below Vmin, the NN accuracy starts to being degraded, i.e., Critical region. For instance, we observe that by 50mV voltage decrease the NN accuracy is dropped up to 3.46%. To prevent it, we propose effective fault mitigation techniques that can significantly prevent the accuracy loss up to 0.1% on. Also, thanks to our fault mitigation techniques, Vmin decreases and thus, the NN accuracy starts to be degraded in lower voltages of up to 30mV. Finally, by further voltage underscaling, the FPGA system crashes with no response at Vcrash, i.e., Crash region. Note that by experimenting on several representative FPGAs, we evaluate the FPGA-to-FPGA variation and observe that the different voltages, i.e., V1stfault, Vmin, and Vcrash differ slightly; however, the fault rate and in turn, the NN accuracy loss in the Critical region is significant, see Figure 11. This variation can be the consequence of the process variation, architectural differences, or aging effect.

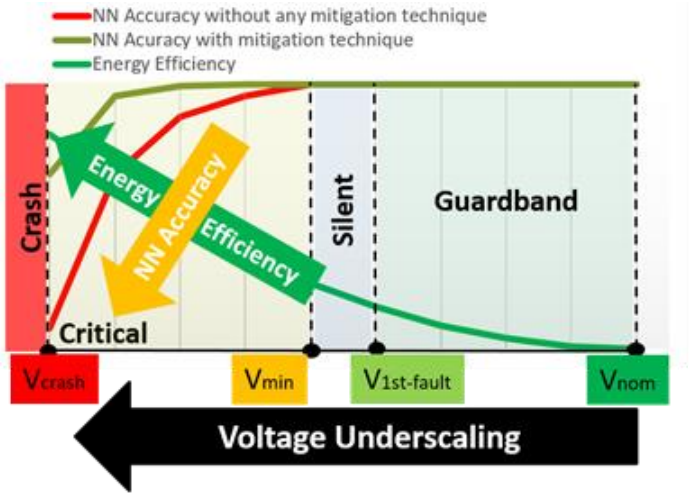
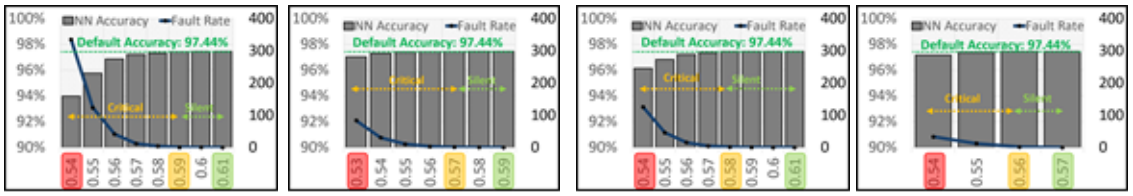


Figure 10 The overall energy-accuracy trade-off via voltage underscaling in a FPGA-based NN.



a. VC707 b. ZC702 c. KC705-A d. KC705-B

Figure 11: Resilience behavior of the FPGA-based NN on four studied FPGAs (x-axis: supply voltage (V), y-axisL: NN inference error rate (percentage), y-axisR: fault rate (per 1Mb), shown for Silent [V1stfault; Vmin) and Critical [Vmin; Vcrash) regions.

+ V1stfault, Vmin, and Vcrash are highlighted.

+ Among different platforms, a slight variation of the voltage regions and the subsequent significant impact on the fault rate and NN accuracy in the Critical region can be seen.

Checkpointing

During these months we have defined the user API to use the checkpoint library. The main objective is to provide the same API for GPU and CPU checkpointing. This will reduce the programmers effort to familiarize with the API as the underlying hardware will be orthogonal to the checkpointing invocation. We have extended the multi-level checkpoint library called Fault Tolerance Interface (FTI) to support checkpoint of data in multi-GPU/multi-node systems. The implementation includes a memory manager, which automatically tracks the physical memory location of user defined virtual addresses. The functionality transparently handles CPU, GPU as well as unified memory addresses. Upon a checkpoint invocation the memory-manager tracks the actual location of the data to be stored, and stores the data accordingly to the stable storage. We profile and optimize the checkpoint procedure. To be more precise, during the C/R, we use two streaming channels, the first is responsible to store the data to the C/R file, and utilizes the CPU resources. The second channel, uses the GPU resources to compute the checkpoint file integrity checksums. To reduce the overhead, both

channels use streams, and are executed in parallel, to move the data from the GPU to the CPU and vice versa and consequently they overlap their operation.

We have also implemented and improved differential checkpointing for GPUs with FTI. In this version of differential checkpointing, a hash function is used to detect parts of the data that change and others that do not change to only checkpoint the block that have changed. This minimizes network contention and parallel file system usage. In addition, FTI makes use of the GPU to quickly compute the hashes and stream the data in parallel with the I/O storage access. This differential checkpointing feature has been accepted and presented at the 19th Annual IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing (CCGrid 2019) and a version of the paper is available on green access at https://arxiv.org/abs/1906.05038.

Moreover, we have gone beyond our initially planned work and we have explored how a thread-based implementation can achieve transparent application level checkpointing with sufficient runtime collaboration. This work helps to maintain the network performance during checkpoints and it also allows for the oversubscription of threads to transparently perform the checkpoint data replication in the background thanks to a dedicated user-level scheduler support. This work has been published in the Journal on Parallel Computing, a version of the paper is available in green access at https://arxiv.org/abs/1906.05020.

Security

Runtime security aspects are discussed together with compiler-level approaches in section T4.7

1.2.4 Work Package 4 – Tool-chain Front End

LEGaTO's work package structure is inspired by the project's software organisation, as a stack of abstraction layers. At the lowest layer we develop the hardware, at the highest layer we offer applications. Work Package 4 focuses on higher-level abstractions, with a programming interface to support the compilation and the execution of secure, resilient and energy efficient applications. This section summarises the technical progress done in Work Package 4, during the first 18 months of LEGaTO.

Work Package 4 offers components for compiling and executing applications with high productivity, performance, security and fault tolerance. When designing a global architecture encompassing the entire LEGaTO tool chain to support applications, we chose to split them into front-end and back-end components. The front-end components are developed in WP4, while the back-end ones are developed in Work Package 3. Some components may have parts both in the back-end and in the front-end of the tool chain, and for the sake of clarity and coherence, we decided to wholly present them either in Work Package 3 or in Work Package 4.

From inception, the development of building blocks in Work Package 4 was split into six tasks: design, programming model, IDE plugin, compiler support, high-level synthesis, task-based mapping, fault-tolerance and security. Task T4.1 (design) has finished, while all other tasks presented good progress in the first reporting period. More detail is given for every task in specific sections of text, below. The work done has been published (or is ready to be published) in selected relevant scientific venues.

The first deliverable of Work Package 4, Deliverable D4.1, **Definition/Design of the Front-End Toolbox**, was finalised in month 9 and included in Superdeliverable SD1 as its Chapter 4**.** It presented a comprehensive set of functionalities designed to be offered as front-end

tools for programming applications in LEGaTO. It specified a software architecture introducing all main aspects on which the project is focused as fault-tolerance, heterogeneity, multicomputer execution and energy efficiency. It detailed the programming models to be developed, extended and integrated in the context of the project. It also presented number of extensions to be implemented in the infrastructure management (OpenStack) in order to support the execution of the proposed task model.

Work Package 4 ended its first reporting period during the preparation of Deliverable D4.2, **First release of energy-efficient, secure, resilient task-based programming model and compiler extensions**, due month 20. D4.2 will present an integrated catalogue of the components implemented in the first 18 months of Work Package 4.

Task 4.1: Definition / Design

Task 4.1 was devoted to producing Deliverable D4.1 (Chapter 4 in SD1) and ended at month 9. It concentrated its activity in designing the functionalities to be developed in LEGaTO's Work Package 4. These tools offer a programming interface for LEGaTO's use case applications developed in Work Package 5 and will be supported by a correspondent runtime system developed in Work Package 3. Figure 2 (extracted from D4.1) depicts the overall software toolchain for the cluster runtime. The LEGaTO programming model front-end is shown on the left-hand side of the figure. The LEGaTO front-end consists of the tools that process the source code and generate the LEGaTO binary targeting the heterogeneous platforms. These tools include extensions to Mercurium (previously developed by BSC) to analyze OmpSs source code and generate Nanos/XiTAO/FPGA/GPU binaries, and two high level programming methodologies to generate dataflow kernels: DFiant and MaxJ.

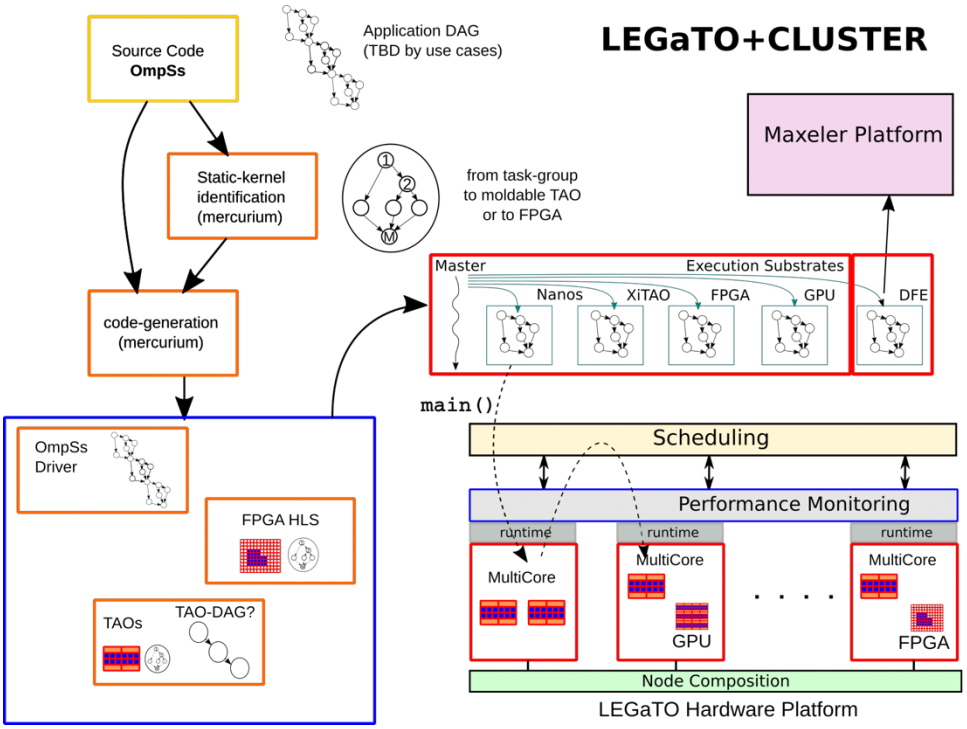


Figure 12: The overall software toolchain for LEGaTO, presented in Deliverable D4.1.

Task 4.2: Programming Model Extensions for Energy Eficiency

LEGaTO toolchain, and the development of programming model extensions for LEGaTO targeting energy-awareness. The structure of LEGaTO's programming and execution model is summarized in Figure 13.

Programming model

Work on the first objective has been well conducted. In order to effectively tackle the second objective, the project partners need to collect experience from the first implementation of the LEGaTO use cases on the LEGaTO runtimes and hardware. This experience will then drive the development of programming model extensions.

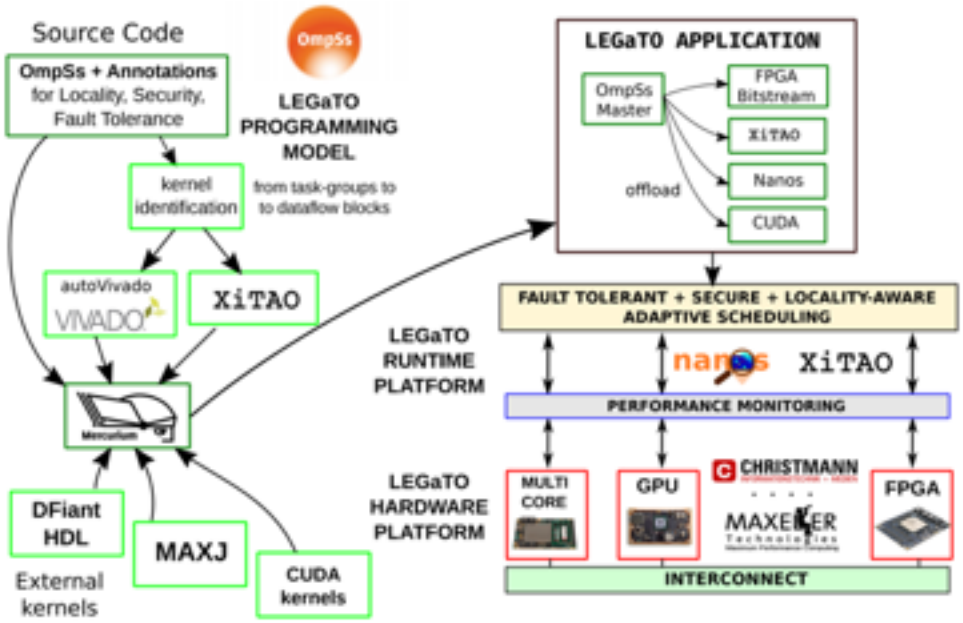


Figure 13: LEGaTO programming and execution models

In the context of the overall execution model, we have so far defined the overall software development flow and execution model for LEGaTO applications. On the programming side, the LEGaTO toolchain will be driven by the OmpSs programming model. LEGaTO applications start as a single thread (called master) which over time generates new tasks that can be executed in by the different hardware resources considered in LEGaTO: CPUs, GPUs and FPGAs. Tasks to be executed on these devices can be natively generated from OmpSs code for CPU cores (Nanos, XiTAO) and FPGA (AutoVivado), or as externally provided kernels for the CPU cores (XiTAO), GPUs (CUDA, OpenCL) or FPGAs (DFiant HDL, MaxJ). Depending on the task type, a particular runtime can be selected for its execution. In the particular case of Nanos and XiTAO, this requires partitioning the CPU cores between the two runtimes. This will be achieved by statically allocating a subset of resources from the master and booting the XiTAO runtime on this subset of cores. Once this step is completed, a XiTAO program is executed on this subset of cores. This development is currently being conducted in close interaction between the XiTAO and Nanos teams, and is expected to yield a first working implementation in the coming weeks.

Annotations

We advanced the implementation of fault tolerance in the front-end toolchain through explicitly annotating tasks for checkpointing, as well as statically annotating the most reliability-critical tasks for selective and energy-efficient replication. The checkpointing

annotations will be propagated to the runtime, which will dynamically determine the exact memory working set of each task before it is fired for execution. This information will be exploited to develop an energy-efficient checkpointing scheme that checkpoints just the data that the task is going to use, instead of the complete memory state. The reliability criticality annotations will also be leveraged to save energy, e.g., the non-critical tasks can run approximate low-energy versions for minimal impact to the results (e.g., in the machine learning use-case that largely relies on heuristics and statistical methods).

Implementation of security measures in the front-end compiler on the bases of annotation in the source code and their exploitation at runtime will be in principle similar as for fault-tolerance. Specific tasks can be associated with particular security requirements and the security measures will be applied at runtime.

Energy efficiency

We have been working on analysing the energy-efficiency and security with different types of hardware. The security analysis was formally reported as part of the paper “Security, Performance and Energy Trade-off of Hardware-assisted Memory Protection Mechanisms” published at SRDS’18. In the paper we reported the energy consumption of Intel SGX and AMD SEV under several workloads for micro- and macro-benchmarks.

The analysis on heterogeneous hardware was used as a base for the design of a task-based scheduler called HEATS. The scheduler was implemented in Python on top of the Kubernetes API and tested on a heterogeneous cluster consisting of 4 different configuration of hardware and using synthetic as well as real world traces. This work has been formally reported as a paper called “HEATS: Heterogeneity- and Energy-Aware Task-based Scheduling” published at PDP’18.

We have further improved HEATS by designing an updated version of it where not only migrations across heterogeneous nodes are exploited but also across the three layers of the deployment architecture (edge, fog and cloud). The prototype implementation and test of the system update is currently under development. On top of that, we have also observed performance and energy improvements when tuning the CPU frequency of the nodes. Based on this, we have developed a sprinting approach which runs jobs at a higher frequency for as long as a predefined budget is not used up. In this work we also take into account that jobs might have different priorities. The system was implemented in Go and tested using Spark. The outcome of this work has been submitted to Middleware’19 in a paper called “Differential Approximation and Sprinting for Multi-Priority Big Data Engines” which is currently under revision.

Finally, we worked on a more extensive evaluation of the ARM TEE, TrustZone. This is a more in-depth performance- and energy-wise study of TrustZone using the OP-TEE framework, including secure storage and the cost of switching between secure and unsecure worlds, measuring emulated and actual hardware. The results provide interesting insights on the energy trade-offs induced by the functionalities used by trusted applications. This study, titled “On the performance of ARM TrustZone” was published and presented at DAIS 2019, 14th IFIP International Conference on Distributed Applications and Interoperable Systems, held in Copenhagen (Denmark).

Task 4.3: IDE plugin

As part of Task 4.3, we incorporated OpenMP and OmpSs support into Eclipse. We developed two supporting plugins to include both sets of directives, including parallel, work sharing, and tasking. These plugins work during application writing, and with them, the Eclipse editor is capable of provide suggestions for directives and clauses while the programmer is typing. For each directive, only the valid set of clauses is suggested. Parameters that can be provided to clauses are also included in the suggestions. For the OmpSs case, we also included the target directive, with the suggestions for the device clauses. The OmpSs plugin also provides the possibility to invoke the OmpSs compilation system, including the invocation of the Mercurium compiler for compiling OmpSs@FPGA aplications, with autoVivado, thus getting the final binary for the host cores, and bitstream for the FPGA, automatically.

During the development of this task, we observed that a new version of Eclipse had been released, Eclipse CHE, and we adapted the OpenMP and OmpSs plugins to it. Eclipse CHE has the advantage to work with docker, in such a way, that we can provide a docker container with OmpSs@FPGA installed, and Eclipse CHE automatically uses it as the development environment. This way, we also provide a preinstalled version of the tools, and the programmer can use them without having to install them previously.

Task 4.4: Compiler support

During the first reporting period for Task 4.4, we improved the support of the OmpSs autoVivado, to compile OmpSs applications targeting Xilinx FPGAs and specifically we have extended its support to discrete Alpha-Data FPGAs, connected through PCIe with Intel boards. This is the first discrete board that OmpSs@FPGA supports. For it, we had to automate the linking phase of our Mercurium generated IP blocks with the application accelerators, with the PCIe IP block provided by the vendor, Alpha Data. The xdma library has been extended to work on top of the PCIe IP block, supporting the data transfers between the host memory and the FPGA-side DRAM memory. The rest of the communication, from the FPGA DRAM memory to the FPGA BRAMs is implemented in the same way as with integrated boards.

The Legato COM Express board is also integrated in the autovivado framework. Through efforts from BSC, we keep also the support for OpenCL updated on Intel FPGAs. Currently Arria 10 is supported with OmpSs@OpenCL.

Task 4.5: High-level Synthesis for FPGA

Prevalent hardware description languages as Verilog or VHDL employ register-transfer level (RTL) as their underlying programming model. A major downside of the RTL model is that it tightly couples design functionality with timing and device constraints. This coupling increases code complexity and yields in code that is more verbose and less portable. **DFiant**, in contrast, is a Scala-embedded HDL that leverages dataflow semantics to decouple functionality from implementation constraints. The main goal of Task 4.5 is the development of the DFiant language and toolchain.

DFiant enables the timing-agnostic and device-agnostic hardware description by using the dataflow firing rule as a logical construct, coupled with modern software language features (e.g., inheritance, polymorphism) and classic HDL traits (e.g., bit-accuracy, input/output ports). Figure14 show how DFiant subsumes traditional HDL languages so it can be used to

write code that is substantially more portable and more compact than the equivalent design in RTL and HLS languages.

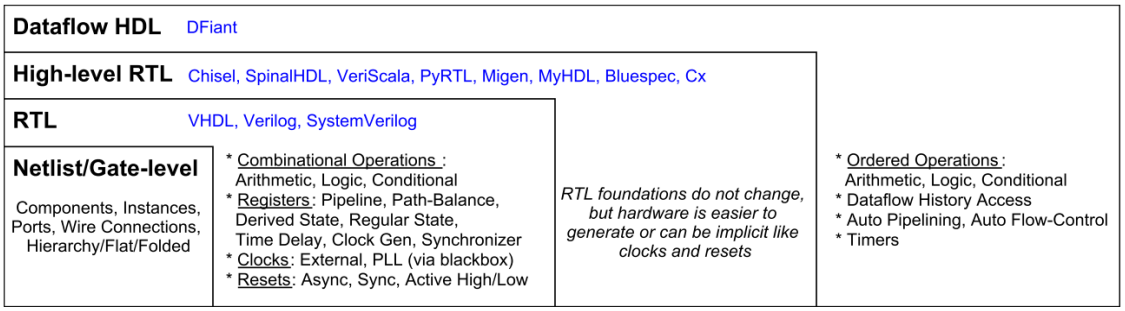


Figure 14: HDL abstraction layer summary (lowest=netlist, highest=dataflow). Each layer subsumes the capabilities of the layer below it. Dataflow constructs replace RTL registers with their true functionality (e.g., state) or inserts them implicitly (e.g., pipelining)

During the first half of the project we have extended the language and toolchain to support many features, as described below. We have implemented both the DFiant compiler and backend. Our compiler backend targets classic synchronous logic and maps the transient state construct into wires and registers.

We extended the DFiant language with the concept of transient values that unify hardware wires and registers. DFiant assumes every dataflow variable is a stream and provides constructs to initialize the token history via the `.init` construct, reuse tokens via the `.prev` construct, and update the state via the `:=` construct. One advantage of the unified representation of transient state is that the DFiant expresses the design structure just like its RTL counterparts, but the resulting code is very concise since state elements are automatically constructed when a stream history is accessed. Another advantage is portability, because state elements are not registers and any type of state component is applicable.

We implemented the DFiant automatic pipelining in the DFiant backend. The backend analyses the design and places registers to split long combinational paths. The compiler has a propagation delay (PD) estimation database that can be tailored for any target device and technology. With this information and a target clock constraint the compiler tags the dataflow graph with the additional pipe stages required before producing the RTL code.

We implemented several common designs and compares their design productivity (in terms of lines of code) and performance to open-source counterparts. These designs include an AES cypher, an IEEE-754 double precision floating point multiplier, a RISC-V single-cycle core, and a bitonic-sort network. Now that the DFiant language and toolchain are more mature, we plan to implement some of the LEGaTO applications using DFiant and to generate FPGA bitstreams.

Task 4.6: Task-based kernel identification / DFE mapping

The purpose of Task T4.6 is to identify static sub-graphs in the OmpSs task graph and map them to kernels on a Maxeler FPGA-based Dataflow Engine (DFE). The rational is that the OmpSs tasks appear naturally suitable for FPGA mapping: They have clearly defined inputs and outputs and have self-contained state. Maxeler's programming model is based on dataflow where large dataflow graphs, described in MaxJ, are mapped and optimised to generate FPGA configurations. These dataflow graphs are essentially static, highly

customised and ultra-deep pipelines that achieve very high computational throughput. Generating these dataflow graphs is supported by Maxeler's MaxCompiler toolchain and runtime execution from a host application is enabled through the MaxelerOS runtime. A task-based programming model such as OmpSs is a good match to act as a front end for the dataflow graph generation. However, due to high context switching overhead of FPGAs, tasks graphs mapped to FPGAs need to be static.

MAX and BSC have begun to collaborate on using the OmpSs programming model as a front-end for dataflow compute kernel generation. First, both partners spent time to familiarise themselves with their respective tool chains. To explore various approaches for OmpSs and MaxJ integration, two practical code examples are used: a tiled matrix multiply example and a convolutional neural network (CNN) that exist in either OmpSs or MaxJ form. Several code integration concepts have already been developed and a key aspect under consideration and development is how to express the number of execution cycles for tasks that are mapped to dataflow kernels. This information is needed for the generation of dataflow kernels and Maxeler currently offers a simple API to dataflow kernels called the “Basic Static SLiC Interface”. This interface is a simple C function with parameters for all incoming and outgoing streams, scalar parameters and the number of cycles each computate kernel has to execute. Here, the automatic mapping from the current notation to the basic static interface is very straight forward with the exception of execution cycles. Current work includes exploring an additional pragma denoting execution cycles to OmpSs which could then be automatically be propagated to MaxJ. This work is on-going and future steps may include a more customised interface to Maxeler's engine interfaces.

Task 4.7: Fault Tolerance and Security

The work carried out within the past 18 months was focused on two aspects: strengthen the platforms protection against security threats and increasing its availability by integrating appropriate fault tolerance mechanisms.

Measures Against Security Threats

To address **side channel-based attacks**, we integrated several protection mechanisms in SCONE's toolchain. For example, the infrastructure was updated with the latest microcode in order to mitigate cache-based side-channel attacks. In order to ensure that applications utilizing Trusted Execution Environments through SCONE do suffer from side channel-based attacks, we extended the attestation service such that prior to the start of an application, the environment is checked to run the latest microcode. In addition to those mechanisms, the SCONE runtime has been also extended to verify the configuration settings in order to prevent any data leakage that might arise due to improper configuration parameters.

Applications utilizing the SCONE runtime require certificates as well as secrets in order to store or transfer data in a secure manner. This requires in turn a **secret management service** that provides full end-to-end encryption such that no human interaction is needed in order to generate or distribute secrets. In order to achieve this, we implemented a secure key management service that has been tightly integrated into the SCONE's toolchain. The key management service is also coupled with the attestation service such that secrets are only distributed and shared with application instances that have been properly attested before. The key management service also comprises a policy board service where only a majority of policy board members can change the configuration such as elevating access rights. This work is currently under submission at USENIX ATC '19.

In order to deal with **untrusted environments** and ensure that applications running in enclaves are only executed in environments along with benign applications, we developed a TPM-based approach that detects and only permits those runs.

Applications running in Trusted Execution Environments such Intel SGX using SCONE often incur a non-negligible overhead. In order to appropriately assess the performance and allow a thorough analysis with regards to performance bottlenecks, we developed a **performance monitoring framework** that has been tightly integrated into the Intel SGX drivers as well as uses performance counters in a way such that applications can be monitored in a transparent manner without source code modifications and no extra developer effort. Furthermore, our approach utilizes state of the art monitoring visualization and storage tools such as Prometheus[5] and Grafana.[6] Using this approach, we assessed the performance of applications running in SCONE as well as Graphene and were able to quantify performance penalties of each of these frameworks.

To assess the applicability of our mechanisms and approaches we proposed, we implemented several **prototypes and demonstrators**. First, we provide applications around secure data analytics such as SGX-PySpark[7] which is also under submission at WWW'19. Besides data analytics, we also assessed the performance of machine learning applications such as Tensorflow. Our secure version SGX-Tensorflow-Lite[8] is like SGX-PySpark publicly availability for evaluation by the community. Furthermore, we provide attestation of cloud services and infrastructure and have created several demos to showcase its applicability.[9] These mechanisms are also described in a paper we recently submitted to USENIX ATC '19.

We also evaluated the usability and performance of **secure services** developed within **ARM TrustZone**. We conducted evaluations on inter-communication between rich execution environment (REE) and trusted execution environments (TEE) using shared memory as well as secure storage. The storage and data exchange technologies were both evaluated by means of micro-benchmarks. In the results of our micro-benchmarks we discovered a significant world switching overhead with ARM TrustZone. The results of our usability and performance evaluation were published in the paper "Developing Secure Services for IoT with OP-TEE: A First Look at Performance and Usability" presented at DAIS 2019, 14th IFIP International Conference on Distributed Applications and Interoperable Systems.

In the work previously described, we identified an **overhead when switch between worlds** (trusted-untrusted) with respect to networking in ARM TrustZone. For the purpose of specifically studying the overhead, we developed a secure service to measure network performance within ARM TrustZone. Among network performance metrics, we also collected energy and timing metrics during our benchmarks to further investigate the world switching overhead. The results on our follow-up work were submitted and are currently under review for MASCOTS 2019, 27th IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems.

Fault Tolerance Mechanisms

In order to improve the availability of the platform we implemented a **secure checkpointing mechanism**. Our first version was implemented using the file system shield available in SCONE. Moreover, we added support for the systems calls `vfork()` and `fork()` in the

[5] https://prometheus.io/
[6] https://grafana.com/
[7] https://github.com/legato-project/sgx-pyspark-demo
[8] http://tinyurl.com/yyfz7qb3
[9] http://tinyurl.com/y6zonpmr

SCONE toolchain in order to provide additional fault tolerance mechanisms such as rejuvenation which is a commonly implemented fault tolerance techniques. Providing fork support for applications running in Trusted Execution Environments is a non-trivial problem. First it requires to create a new enclave and copy the whole application state to the new enclave including running an attestation. Furthermore, the state must be consistent when using multiple enclave threads. It also requires pre-emption of non-forking threads. We have recently completed the implementation of this functionality which is currently under testing at the point of writing of this report.

1.2.5 Work Package 5 – Application development and optimization

WP5 aims to showcase the LEGaTO toolset on five application domains. These application domains correspond to the following tasks of the workpackage: SmartCity (T5.2a), SmartHome (T5.2b), Biomarker discovery (T5.3), Machine Learning (T5.4) and Secure IoT Gateway (T5.5) use cases. Additionally, T5.1 specified the application optimization and evaluation plans. This section summarises the technical progress achieved in WP5 during the first 18 months of LEGaTO.

Task T5.1 has finished, while the progress of all the other tasks are ongoing. For all the ongoing tasks, we summarize the technical progress and provide a list of next steps. In accordance with interim review comments, we decided to put additional emphasis on support for Machine Learning and Smart City/Home use cases in other Work Packages.

Work Package 5 ended its first reporting period during the preparation of Deliverable D5.2, First report on development and optimization of use-cases, due month 20.

Task 5.1: Analyse and specify application optimization and evaluation plans

The purpose of this task was to analyse and define the optimisation paths for each of the use case applications as well as to define the evaluation criteria and metrics for the evaluation of the optimised use cases. This includes a detailed analysis of each application, the description of its algorithm structure and performance aspects. The applications were then matched against the various components of the LEGaTO tool stack and we identified which parts of the tool stack are relevant for the respective applications as well as what the goal of the optimisation is.

It includes the definition of baselines, metrics and evaluation criteria. This task was completed in M9 and the output is covered in section 3 of the combined super-deliverable. Following the reviewer feedback from the first review, this document was improved, providing additional information on specific optimisation goals, details and energy baseline information. This updated version of the super-deliverable has been submitted.

Task 5.2a: Urban-scale air quality model (SmartCity use case)

In many urban areas air quality and associated impacts on public health are matters of growing concern. The emission and dispersion of critical pollutants (PMx, O2 and ground-level O3) correlate with cancer, asthma, cardiorespiratory problems, brain development in children and reduction of life expectancy in general. As a consequence, air quality monitoring networks and modelling forecasting systems are critical to increase awareness and, ultimately, to assist decision-makers on the adoption of measures to protect public health.

In this scenario, the LEGaTO stack is pivotal, both in terms of leveraging the processing capabilities and improving the energy-efficiency of an operational urban-scale air quality modelling system. The Smart City use case aims at demonstrating that monitoring of urban air quality through CFD simulations is feasible for nowcasting predictions in an operational workflow built on top the LEGaTO stack.

Within the Smart City use case, the CFD-based simulator is the main core of the whole application. To this end, an urban-scale wind forecasting system was developed, based on coupling the meso-scale WRF model with BSC's Alya modelling system. On top of this model, the most compute-intensive kernels are being ported to take advantage of the LEGaTO platform. Two main hotspots have been identified as targets: the elemental assembly for the explicit momentum equations, and the Sparse Matrix Vector operation (SpMV) used in both explicit and implicit solvers. The porting to the LEGaTO stack requires at least two operations: the kernel taskification with OmpSs programming model, and the full rewrite of the Fortran90 kernel code to C language.

So far, we have fully ported a basic implementation of the explicit momentum equation to OmpSs and C code. The kernel has successfully been compiled and run on ARM64 platforms (Cavium Thunder and AXIOM board), where the baseline test-case for the SmartCity use case has been conducted. This kernel has also been annotated with OmpSs@FPGA directives in order to run on the Xilinx Zinq Ultrascale+ FPGA on the AXIOM board. However, some work needs to be done as Xilinx' toolchain is producing wrong numerical results on the FPGA side.

Finally, we have also defined the testset for the SmartCity use case. This testset will be used to evaluate the performance gains (elapsed time and Watts) of the LEGaTO implementation with respect to the original code. The testset was run on a single MareNostrum IV node, and the most important metrics were obtained as a baseline. These metrics will be used in future analysis to evaluate the improvement of the LEGaTO implementation.

Our next steps for this use case will involve:

- Fix Xilinx Vivado-HLS compiler issues with the AutoVivado generated code,
- Insert and tune C code with HLS directives,
- Evaluate the reduction of the floating-point precision,
- Porting of the SpMV kernel to C code and to OmpSs tasks.

Task 5.2b: Smart Mirror as a Universal User Interaction Interface (SmartHome use case)

The smart mirror is a more and more frequently used interface for interaction in smart home environments. It is based on a display with a semi-transparent foil applied on it. This device shows personalized information and enables controlling of other smart components and services, e.g., operating the automated wardrobe, turning on/off the lights or opening/closing the entrance door. For this use case a demonstrator based on the open source project MagicMirror² is developed and extended with the most needed features of smart homes. These include face recognition, object recognition as well as voice and gesture control.

Main goal of use case is the reduction of the high power consumption and hardware requirements. The development started with a fully equipped high end PC with two powerful Nvidia graphics cards (GTX 1080 Ti). In the second prototype tensor cores on two Nvidia RTX 2070 were utilized. By this step the initial power consumption of 650 Watt was reduced by 140 Watt showing the potential of the tensor cores. As these are also available in the Nvidia Jetson Xavier they will be part of the aimed edge hardware architecture. Multiple

embedded hardware architectures are evaluated for a further improvement of performance and resource efficiency. These include Nvidia TX2, Nvidia Xavier, Intel neural compute stick and a selection of GPUs. In the scope of the micro server, developed within this project, in a next step we will port the application onto two Nvidia Xavier coupled via PCIe.

Several software modules written in Python were ported to C++ with additional CUDA extensions. By this, the performance of the detection modules (face, object, and gesture) could be increased from around 16 FPS to 24 FPS with all modules running in parallel.

The LEGaTO stack is applied to the object detection module first, which is based on the neural network YOLO (You look only once). The corresponding framework for YOLO is called darknet and written in C. This first step of the evaluation and optimization of this framework is still ongoing. In addition to the object detection, YOLO is also used for gesture detection. For an easy to use gesture control, a dataset consisting of 32 hand gestures was gathered. This dataset contains images of around 20 persons with around 2000 images of each gesture. A YOLO network trained with this dataset is used to control the smart mirror.

A recently added feature is tracking of every detection of faces, objects or gestures. Based on a Kalman filter, this allows for an overall increased user experience. In combination of all tracked detection and recognition information, users can be tracked and are still recognized even when their faces are momentarily turned away. The implemented Kalman filters are currently not optimized and have a high resource overhead. In a next step they can be optimized by the LEGaTO toolflow. All neural networks in this demonstrator also require scaling down the camera image resolution. Due to the structure of the used neural networks and the recognized or identified image patterns, a unified down scaled image is not favoured. The LEGaTO toolflow can be utilized to find a more energy efficient solution.

For the development of a behaviour prediction module, first design decisions were made. The general structure of the communication between all included modules are heavily changed for a wider range of information available for neural network. A first simple RNN structure is currently evaluated. For an efficient training of this RNN, a situation memory will be designed.

Our next steps for this use case will include:

- Porting the smart mirror onto embedded hardware (edge server)
- Optimization of darknet (YOLO) with OmpSs
- Analysis and optimisation of image scaling/Kalman filters
- Immersion of behaviour prediction RNN

Task 5.3: Biomarker Discovery in infection research

To avoid wasting time and money, researchers do pilot studies with a small number of observations as necessary first step for biomarker discovery. Because of the small number of cases and the large number of biomarker candidates, any result might be caused by random effects and statistical significance cannot be proven. For that, the researchers try to reduce the number of biomarkers and extract the most informative ones from these pilot studies to then increase the sample size and achieve an adequate statistical power.

In our first algorithm until now we evaluated the biomarker candidates based on the area under the Roc curve (AUC). This evaluation does not need any simulations, because there is a closed form solution for the probability (p-values for the significance of the biomarkers). We find the best combination of biomarkers to classify the diagnosis and calculate their joint

AUC value. However, because the computational time for combining for example 5000 from 50,000 possible biomarkers (10%) is estimated to be months or years, we could not involve a large number of biomarkers and instead we choose a few of them (<19) with the highest AUC.

We are now ready to use other biomarker performance measures like entropy and misclassification rate. We need to calculate the probability for each biomarker in the real data showing a high value of performance measurement just by chance. We estimate the probability for a specific performance value to occur just by chance. Therefore, we calculate the number of biomarkers in random data whose performance value is equal to or greater than this specific performance value. Dividing this number of biomarkers by the total number of biomarkers in random data we get the probability of the performance of that biomarker.

To measure the probability of the performance values of these thousands of biomarker candidates can take days in the normal computer while in cooperation with Maxeler and using their DFEs, the computing time is reduced dramatically. The application was significantly improved in terms of its performance by porting the performance critical parts from R to MaxJ, targeting a Maxeler DFE. The original 10^6 simulations took 4514 s (1.25h) to run in R. An optimised intermediate version in C ran in 75s and the final DFE-accelerated version of the same problem ran in 5.5s, a speed-up of over 800x compared to the original R code.

We are about to look at more than 50,000 biomarkers so we need to estimate a probability smaller than 1/50000, which means we need to run up to 50 million simulations which can only be done with Maxelers DFEs.

We have developed a second algorithm to discover biomarker combinations by regression and classification. The proof of concept and a validation with small sets of artificial data with about 30 biomarker candidates has been done. Because this approach is even more computationally expensive, being able to apply it to real biological data with up to 50,000 biomarker candidates extreme powerful hardware is needed. The core of the algorithm (Microsoft/lightGBM with MIT licence), a well known machine learning tool, is now implemented with OpenMP. The conversion from OpenMP to OmpSs has started in order to run the algorithm on the hardware provided and supported by the LEGaTO project.

Task 5.4: Machine Learning use case

Although there are many success stories of deep learning (DL), which has spurred a wave of public and corporate interest in AI, there are still many unsolved issues. One of these is how to make the DL architectures more efficient, especially to be used on embedded devices. To solve this problem, we are developing a new kind of DL optimisation tool (EmbeDL) that we will demonstrate on a vision application relevant for autonomous driving applications.

EmbeDL relies on multiple techniques and we have more techniques on the roadmap for 2019 and 2020. A core piece of our technology stack is the hardware aware hyperparameter optimiser that combines different optimisation techniques and parameters, e.g. on a per layer basis set the pruning level. This is vital when the number of techniques and corresponding parameters grow in size and makes it infeasible to combine methods by hand. This technology is useful now, but will be of paramount importance going forward adding more techniques and methods for optimisation.

The models we have used for our benchmark are VGG16 (https://arxiv.org/abs/1409.1556), GoogleNet (https://ai.google/research/pubs/pub43022) and ResNet

(https://arxiv.org/abs/1512.03385). These models are commonly used as the main component in basically all image/video related applications.

We did benchmarks of three common vision models run on Nvidia's Jetson TX2 platform. The result can be seen in the table below.

Nvidia TX2	VGG16	GoogleNet	ResNet
Speed	16x	5x	3.4x
Energy	16x	6.2x	2.4x

VGG16 is the model we have analysed more thoroughly when large parts of the optimiser have been developed. GoogleNet and ResNet were implemented, optimised and analysed after that and improvements to the optimiser were done. We are quite confident that there are some low hanging fruits for improvements.

Furthermore, the integration of OmpSs has started and a first implementation of an OmpSs taskified artificial neural network model has been done for CPU and this implementation has also been ported to OmpSs@FPGA.

Future project goals involve

- Extensions and improvements to the EmbeDL optimisation toolchain.
- Analysis and optimisation of the OmpSs@FPGA integration.
- Implementation and testing of object detection and semantic segmentation tasks relevant to the autonomous driving domain.
- Benchmarks in the legato testbeds.

Task 5.5: Secure IoT Gateway use case

The Secure IoT Gateway is a new development for an easy to use and configure multi-customer data-center ready VPN solution, designed to support a variety of different use cases. It is based on existing firewall distributions for server and edge devices and bundles these resources to an overall managed product.

The Secure IoT Gateway is composed of four components that work together: (1) The Network Cockpit is an administration user interface for monitoring and setting the VPN settings and firewall rules of/between all components. The (2) Gateway Cluster is a multi-customer VPN server that runs in the data center and separates the traffic of all customers, allowing a fine-grained control of the customer's sensible data. The (3) Network Gateway is the central VPN gateway hosted at the customer's premises and secures all traffic to/from the internet to the Gateway Cluster in the data center. Finally, there are many (4) IoT Bridges at the customer, securing the traffic between the customer's Network Gateway and any IoT devices.

The Gateway Cluster and the Network Gateway are full-featured x86 servers with multiple hardware encryption units for a fast openVPN and based on OPNsense, while the small IoT Bridges are powered by openWrt.

The Network Cockpit contains a full user management with two major roles, the data center manager and the individual customer. The whole application supports internationalization. The Gateway Cluster and all Network Gateways are monitored and shown with their health status and VPN traffic. It's also possible to see the connected IoT Bridges and the IoT devices. Some measurements like the data transfer are visualised with statistics.

For setting up the VPN connection between the Gateway Cluster and Network Gateway respectively the Network Gateway and IoT Bridges, an OPNsense plugin is in the planning stage. With this plugin, it will be possible to read and write settings so the user just needs to operate with the Network Cockpit.

The IoT Bridge ensures the connection between an IoT device and the Network Gateway with encryption on-the-fly. To test the VPN connections among the Gateway Cluster, Network Gateway and the IoT Bridge, a testbed containing adequate hardware and operating systems was prepared. This will also be used for implementing and testing the OPNsense plugin.

In order to rate the choice of technology to implement a secure and effective VPN, a benchmark comparison between WireGuard and OpenVPN was made, using server and IoT hardware. These technologies use different encryption methods for the data transfer. Out of the box, Wireguard is faster and much easier to install than OpenVPN but OpenVPN can be configured to use the AES hardware encryption units of modern CPUs and thus wins the race in terms of performance and energy efficiency.

Once the Secure IoT Gateway is ready for a beta test, it will be deployed to UNIBI's smart home to test the integration with real-live workloads.

1.2.6 Work Package 6 – Project Dissemination and Exploitation

The main objective of the project dissemination and exploitation work package is to maximize the impact of the project through various channels.

Task 6.1: Dissemination and communication planning and tools

Since the project started, the dissemination team defined a project branding (logo and brand guide). A flyer and a poster template was developed and is available to be used for all project partners.

The second main task of WP6 in the first project months was the development of the **LEGaTO website** to inform stakeholders and potential audiences about the project and its activities. The website, which went live in February 2018, uses the Drupal platform for content development and management. Google Analytics is being used to monitor web statistics. The website is updated regularly with news articles, software components, use cases, events and publications. Following the review comments after the first review, two dedicated LEGaTO social media channels have been created: SlideShare and LinkedIn. The **SlideShare** account was created in March 2019 so that researchers may be able to share their PowerPoint presentations, posters or other documents publicly. As of M18, it hosted five presentations. A **LinkedIn** account was created on April 1, 2019. As of M18, it had 22 followers. In addition, the dissemination team created a social media plan in order to increase the followers on both LEGaTO dedicated social media networks.

In the first 18 months of the project, LEGaTO partners have attended a total of 19 conferences and meetings and have organized four trainings and tutorials.

There are currently 14 scientific publications which include journal articles, conference proceedings. Publications are always updated on the website on page https://legato-project.eu/publications.

Task 6.2: Exploitation

The exploitation task started from month seven. To collect information on initial exploitation plan and to get a clear view on target markets an exploitation questionnaire was sent to the partners. Each partner was responsible to provide the information. This template helped the researchers to think about the exploitation plan and activities should be conducted under this work package. After reviewing the filled up questionnaires the potential target markets were identified.

In order to keep track on the outcomes from the projects three repositories have been created. The repositories are to collect information on software components, IP and exploitation activities. The repositories are live document, so they will be updating till the end of the project. The detailed Intellectual Property Rights (IPR) and Knowledge Management guideline has been provided in the revised D1.1 (Project management and quality guideline).

Apart from participating in the monthly meeting and face to face (F2F), individual teleconferences have been conducted with each of the partners to provide guidance on exploitation. Each meeting took place minimum thirty minutes to maximum one hour. This supported the researchers to understand and think about the exploitation plan, activities, reporting to the repositories, think about the end users group as well as plan for the standardization activities and collaboration opportunities. The preliminary outcomes of the exploitation questionnaires will be provided as part of D6.3 deliverables.

In addition, under exploitation the coordination and organization of advisory board has been done. The detailed plan has been included in the revised D1.1. According to the plan the first Industrial Advisory Board (IAB) meeting has been conducted on month 17th during the F2F meeting at Tel Aviv, Israel. Three of the IAB members representing the HW experts were invited. The summary of the main outcomes of the meeting, including the meeting minutes will be provided as part of the next deliverable D6.3. It will also include the further developed plan for next IAB meetings.

1.3 Impact

The project is in line with the section 2.1 of the DoA. The detailed individual exploitation plan with early exploitation results will be included in the D6.3, exploitation plan deliverable. However, the summary of the exploitation results has been included below:

From the beginning of the project until M18 15 software components have already been spotted as potential exploitable assets. Among them three are proprietary and rest of them are open source. An early success story within LEGaTO project is the creation of a spinoff company. It can be instrumental for the commercialization of some of the technologies developed within the project. As the technologies under LEGaTO project have potential impact to the emerging low power computing systems scientific community, a lot of companies have showed interest on different components of LEGaTO. Towards the end of the project we plan to expand the scope and reach out other adjacent communities such as Artificial Intelligence, Edge computing, Machine Learning etc. for potential collaborations. The details results and plan will be provided in D6.3.

Following the Key Performance Indicators defined in the Dissemination Plan (D6.1), the table below summarizes the status of the dissemination activities:

Key Performance Indicators	Explanation	Total Target (by the end of the project)	M 18 status
Press releases	At least 1 per year	2	1
Media clippings	Articles appeared in the press about LEGaTO	20	7
Project flyer	At least one brochure regularly updated	1	1
Website sessions	Number of sessions registered by Google Analytics	1,000 sessions/year	12,152
Events and conferences attended	Where the project had a presence and was disseminated through a presentation, booth, poster, etc.	10	19
Scientific publications	Peer-reviewed journals, conference proceedings, etc. – in green open access.	20	14

2. Update of the plan for exploitation and dissemination of result

Include in this section whether the plan for exploitation and dissemination of results as described in the DoA needs to be updated and give details.

A dissemination plan has been submitted at the beginning of the project, and has been regularly updated. It has been correctly implemented. The communication actions such as attending scientific events, updating the website, creating content to disseminate the LEGaTO progress are tasks that have been fulfilled during the 18 project months.

The exploitation of the project is in line with the DoA. Early results are presented part of D6.3 “Exploitation Plan”.

3. Follow-up of recommendations and comments from previous review(s)

The list of the recommendations received in the Interim Review and the actions developed to tackle them were added as an annex in the resubmission of the "D1.1 Project Management and Quality Guidelines". All the recommendations are also attached again in this document in the Annex I "Addressed Recommendations".

4. Deviations from Annex 1 and Annex 2

In July 2018, an amendment was presented to tackle the following 6 points:

1. "Technion Research & Development Foundation Ltd." was added as third party of the TECHNION partner. This is an in-kind contribution estimated in 202.726 € without any impact on the budget.

2. TECHNION increased their number of person-months because they made a mistake in their initial person-month rate. This had any impact on the budget.

9 / TECHNION	WP1	WP2	WP3	WP4	WP5	WP6	TOTAL
Original PMs	1	1	5	40	5	2	54
Proposed PMs	1	1	**8**	**66**	**6**	2	**84**

3. TECHNION: Transfer 1.500 € from equipment to travel.

4. HZI: Transfer 6.000 € from audit costs to personnel costs. This supposed an increase of 1 PM in WP5.

5. Correction of a typo in Task 4.2.

6. Merging of deliverables: The contents of the deliverables D2.1, D3.1, D4.1, and D5.1 were consolidated in order to reflect the tightly-integrated nature of the entire LEGaTO software/hardware stack and to present the entire project, in a more coherent and concise manner. All the contents were merged in a newer version of D2.1 while the rest were deleted.

This amendment was accepted and all the changes made.

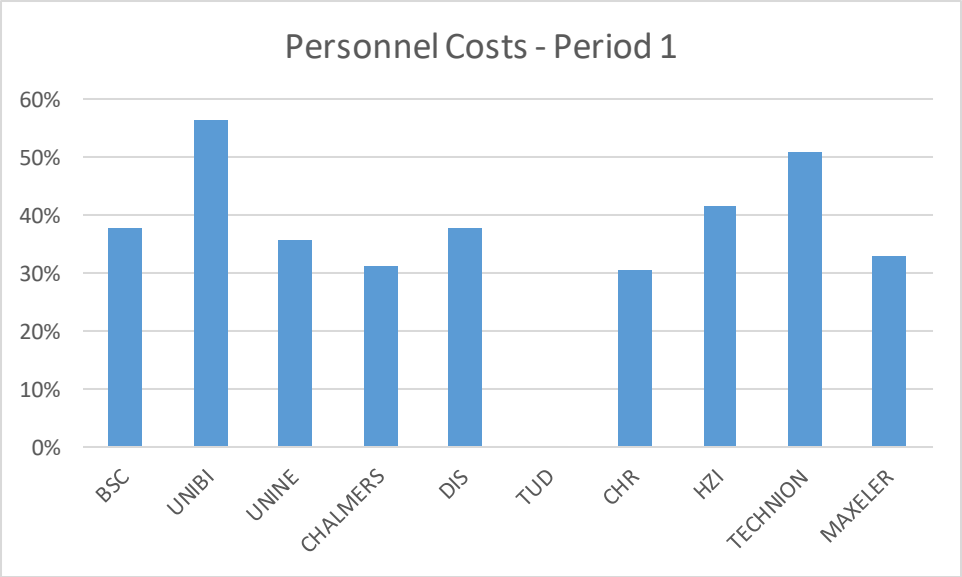
5. Reporting

In this section it is explained the use of resources by all the partners during the first period of the project (M1-M18). These numbers are estimations in some cases and could change in the final Perioric Report to be submitted the 31st of July.

In general, we can see the costs and efforts reported during the first period are below the 50% expected by a linear distribution, but it is expected to be devoted more resources during the second period of the project. Also regarding the Other Direct Costs, it is expected to have a final workshop at the end of the project that will suppose a significant part of the budget.

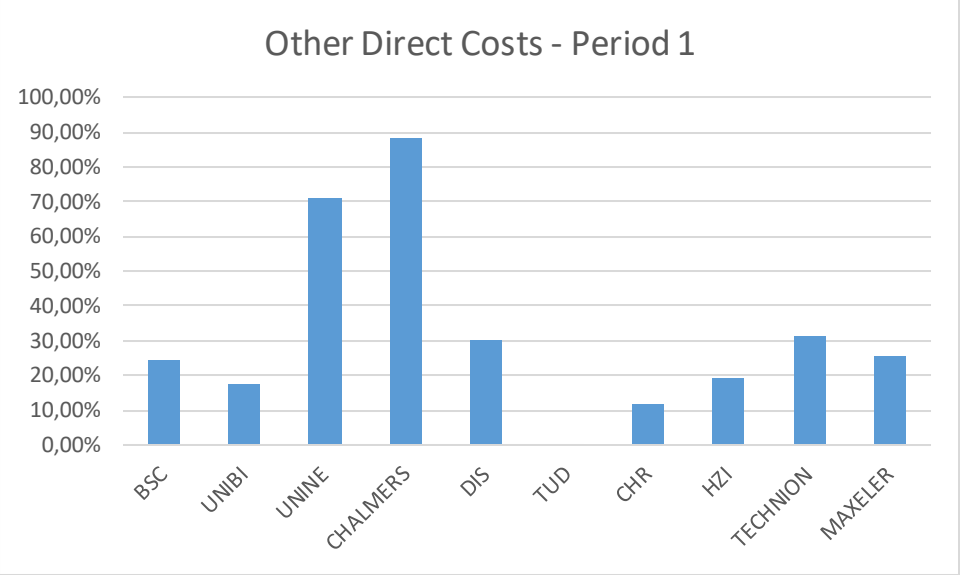
5.1 Personnel costs

Partner	Direct Personnel Costs	Estimation Period 1	% reported
BSC	725.420,00 €	273.863,99 €	37,75%
UNIBI	457.650,00 €	257.920,46 €	56,36%
UNINE	336.000,00 €	120.337,39 €	35,81%
CHALMERS	489.483,00 €	153.531,49 €	31,37%
DIS	114.000,00 €	43.106,00 €	37,81%
TUD	319.200,00 €	-	-
CHR	558.600,00 €	170.474,59 €	30,52%
HZI	137.400,00 €	57.459,96 €	41,82%
TECHNION	324.000,00 €	165.580,57 €	51,11%
MAXELER	440.000,00 €	145.693,80 €	33,11%
	3.901.753,00 €	**1.387.968,25 €**	**35,57%**



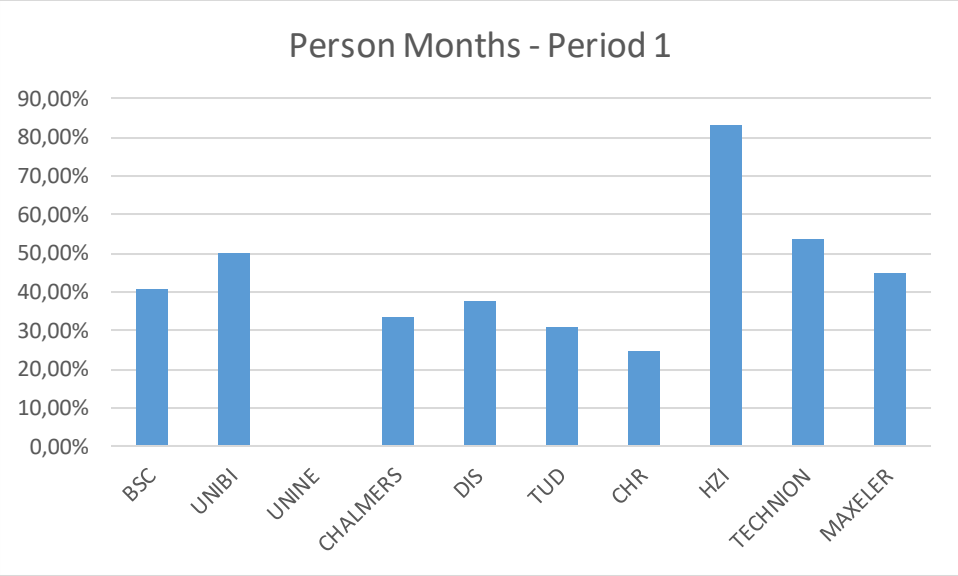
5.2 Other Direct Costs

Partner	Other Direct Costs	Estimation Period 1	% reported
BSC	145.642,50 €	35.777,85 €	24,57%
UNIBI	65.700,00 €	11.656,80 €	17,74%
UNINE	25.700,00 €	18.209,82 €	70,86%
CHALMERS	33.850,00 €	29.825,59 €	88,11%
DIS	28.200,00 €	8.536,00 €	30,27%
TUD	25.700,00 €	-	-
CHR	89.200,00 €	10.375,84 €	11,63%
HZI	29.200,00 €	5.530,53 €	18,94%
TECHNION	38.200,00 €	11.882,20 €	31,11%
MAXELER	28.200,00 €	7.188,98 €	25,49%
	509.592,50 €	**138.983,61 €**	**27,27%**



5.3 Person Months

Partner	Person Months	Estimation Period 1	% reported
BSC	166	67,51	40,67%
UNIBI	81	40,5	50,00%
UNINE	42	-	-
CHALMERS	81	26,99	33,32%
DIS	20	7,5	37,50%
TUD	56	17,38	31,04%
CHR	98	24,42	24,92%
HZI	24	20	83,33%
TECHNION	54	29,12	53,93%
MAXELER	55	24,8	45,09%
	677	**258,22**	**38,14%**



Annex I: Addressed Recommendations

Recom. No	Reviewer recommendations:	Action to be taken/Implemented changes
R1	Review all deliverables for basic formatting and quality. All deliverables should have a clearly separate Executive Summary of 1-1.5, a distinct Introduction and a specific Summary or Conclusion section. The project has developed nice branding and colour schemes; these could be considered for use on e.g. the first page of the deliverable or in the header/footer etc. as a means to make the deliverable look more attractive.	For each deliverable (and each chapter in SD), included executive summary, introduction and summary. Improved the format for deliverables. The deliverables template has been updated.
R2	The project needs to quickly clarify the situation concerning the Industrial Advisory Board. While it is recognized that some efforts have been made in this direction, it is not specifically clear how and when the IAB will engage with the project and it is not clear that they will be able to provide useful input.	- Our first IAB meeting took place on 9th April, 2019 at Tel Aviv during the f2f meeting. - We are going to provide a feedback report in D6.3, which is due in M20 - Considering the comments from the reiewers we are now trying to involve more Industrial end users group to our advisory board - So we are renaming it as Industrial and End Users Advisory board The detailed description and plan is included in resubmitted D1.1
R3	The project needs to make its Open Source outputs clearly visible from its web page. It is understood that the project is leveraging previous work and some thought may need to be put into giving appropriate credit to previous work while also giving sufficient credit to LEGATO, but this is not difficult in principle. Appropriate pointers to software documentation should be clear and LEGATO should provide some information on how the disparate components can be integrated/used together.	It has been created a whole new section in the website to cover this recommendation: https://legato-project.eu/software-components
R4	Make specific modification to Deliverable D1.1 based on the comments below. Specifically, changes relating to each of the following points are required: (i) Industry Advisory Board, (ii) more fine-grained milestone definition, (iii) innovation and IPR management and (iv) software	(i) Industry Advisory Board: please refer to R2 action, (ii) additional milestone added at M24 (please refer to D1.1) (iii) Innovation and IPR management: the detailed description of the role of the Innovation manager and IPR management has been prepared. Also created two repositories in SVN to collect information from the partners regarding IP and software components (iv) Please refer to R3 action

	management and software quality assurance	
R5	Make specific modifications to Deliverable D6.2 based on the comments below. Specifically, changes relating to new and potential data sets arising from the work are required.	Partial rewriting of the D1.2 following their comments.
R6	Make specific modification to Deliverable D6.1 based on the comments below. Specifically, changes relating to the following are required: (i) different constituencies with which the project should communicate need to be considered more clearly, including appropriate messages for them, (ii) if developers/engineers are one of the primary constituencies the project is targeting this needs to be given due attention	Update of D6.2 done with target audience table included.
R7	Make specific modifications to Deliverable D2.1 based on the comments below. Specifically, changes relating to the following are required: (i) a summary table highlighting how the applications can benefit from LEGaTO, (ii) more details regarding baseline energy consumption for the use case applications, (iii) clarification regarding the purpose of the energy model devised, (iv) the inclusion of data pertaining to the FPGA undervolting work, (v) an alternative TCO calculation assuming lower power consumption per rack.	(i - iii) Discussion of how applications can benefit from LEGaTO has been added, table for energy for use cases including the baseline has been added, the energy model purpose is clarified, and the section moved to runtime chapter (iv) Data from the FPGA undervolting is made publicly available (v) The TCO calculation in chapter 6 for D2.1 has been updated to include an additional scenario with lower power consumption.
R8	The publication related information on the LEGaTO website needs to be amended as noted below. Slide decks, which have been given relating to talks, should also be made available via the project website, perhaps using a LEGaTO slideshare account.	https://www.slideshare.net/legato-project (Publicatoins have also link to the PDF further to the slideshare, e.g.: https://legato-project.eu/publication/comprehensive-evaluation-supply-voltage-underscaling-fpga-chip-memories)
R9	The project needs to more clearly define groups who could use components of the LEGaTO framework and engage with these communities. The OmpSs community is interesting but it is small; developers of SmartHome applications is more likely a larger community and developers	- Meeting with CLASS and ELASTIC projects for potential synergies - LEGATO will try to participate in AI events that will take place in Europe. - Use Cases included on project website: https://legato-project.eu/use-cases

	of neural network based application is also large – the project should review which communities have scale, momentum and focus energies on engagement with these communities.	
R10	A number of important new development platforms have received significant interest in the last year, including RISC-V (including bespoke processors with e.g. neural network inference extensions), Google EdgeTPU, Intel Myriad X-VPU. The project needs to maintain a watching brief on such new platforms and may be able to obtain early access to some of these platforms for experimentation purposes should it be appropriate. It any case, it should perform a lightweight assessment to determine if LEGaTO is well suited to such newer platforms and in particular if there may be 'easy wins'.	Propose an evaluation mechanism for suitablility of LEGaTO technologies on new and upcoming hardware platforms. Include report for next review and for M36. (WP2): The edge chapter has been updated by a note mentioning the ongoing activities in the project wrt. integration of new form factors.
R11	The project needs to ensure an adequate portion of its work/resources targets development platforms, which have some traction or provide solutions to enable work to be ported from widely used development platforms to the LEGATO framework.	Target development platforms such as Eclipse, present proposal at EclipseCon and similar industry meetings..

Objectives and Workplan

R12	WP6 has been progressing promotion of the project, with visibility in 6 media outlets and having 7 publications, which is commendable for the first 9 months of the project. The project has performed some analysis of market opportunities; while this work is interesting, the focus should shift away from considering LEGaTO as an indivisible unit and focus more on smaller components, which may have commercial potential.	Now our market is divided in horizontal and vertical markets focusing on each components of LEGaTo. The detailed report will be provided in D6.3 Exploitation Plan deliverable.
R13	Risk 4 ("Emerging disruptive technology from other suppliers [...]") needs for periodic monitoring along the project duration, but so far, there is not visibility on the results of this monitoring activity or the potential adoption of these technologies by LEGaTO. This is highlighted in recommendation 10.	Propose an evaluation mechanism for suitablility of LEGaTO technologies on new and upcoming hardware platforms. Include report for next review and for M36
R14	The SmartMirror application is compelling and very demonstrable. Even though the base technology has not been developed specifically by the project, the project can highlight its valuable work on making it more energy efficient and easier to work with. The project should leverage the very interactive nature of this demonstration to maximize its marketing potential.	Participating in industry events such as Teratech to promote the smart mirror application
R15	The Smart City use case has not demonstrated innovative results yet. Further, as noted at the review, a baseline for the typical energy consumption of the CFD models must be provided to assess gains delivered by LEGaTO	Added the energy consumption for the baseline version in the deliverable
R16	The Machine Learning use case has provided basic information on a Deep Learning optimization technique, which delivers 4-5x performance over a baseline. However, limited details have been provided neither in the deliverable content nor at the review. We look forward to hearing about progress in this area in more detail at the next review.	MIS will prepare a deep dive presentation with more details for next review
R17	The Infection use case has provided some basic synthetic analysis, which indicates that significant performance gains of almost 3 orders of magnitude could be possible by porting their R code to code running efficiently on the Maxeler DFE engines. It will be interesting to see if such gains can be attained for even smaller variants of the real calculations to be performed by HZI	At the moment we are still developing the algorithm. Therefore we can not perform calcualtions of real data by now.
R18	The secure IoT Gateway encountered issues in the analysis, which meant that the scope for optimization was very limited. The project adapted somewhat by considering how this could be used for securing other applications sitting on top of it.	The secure IoT Gateway will be used to secure the communication of the Smart Home use-case.

Impact

R19	Smart Home/City use cases: The performed adaptation of ML libraries for the SmartHome use case and the TBC library for the SmartCity to OmpSs (as presented during the review session) is a relevant step for these use cases to benefit from the power-reduction capabilities of the LEGaTO framework.	It is being managed to port darknet to OmpSs and will report the progress in the next review.
R20	Healthcare use case: The use cases is focused on computation power allowing to analyse bigger sets with a pre-selected set of hardware components (§C). While this can be seen as reduction in power consumption, to fully get credit from power saving the project should present an estimation of the power consumption of these bigger sets (e.g. projection for these bigger sets based on the power consumption of the current sets using the currently available hardware).	We have simulated the entropy values for only 3 biomarkers 1e6 times for 66 observations and 4 classes . It took 14,897 hours. The estimated energy consumption for this calculation was 1,26kWh. Real datasets have about 50.000 biomarkers and are not calculable yet.
R21	The current status of the LEGaTO framework is in the right path to have an impact on the availability of low-power technologies for non-experts on the field through the use of OmpSs and its annotations (§4.1 D2.1, §4.2 D2.1) and the synthesis of accelerators by means of High-Level Synthesis (HLS) languages (§DFiant §4.1.5 D2.1, MaxJ §4.1.6).	The HZI will port a second application to OmpSs and plans to publish a well known ML algorithm (lightGBM) adapted to OmpS
R22	However, the dissemination activities of the project should also address non-OmpSs users, specifically for ML healthcare user communities that can benefit from the ML libraries adapted to OmpSs. Given the relatively small user-base of OmpSs, this is a concern with respect to the impact that can be realised by the project.	Dissemination and Exploitation teams will take into account this sector to reach them in future activities.
R23	Progress towards this Expected Impact for the members of the consortium is adequate with the three SMEs and mid-caps in the consortium (Data Intelligence, Christmann and Maxeler) increasing their innovation potential through sound technology development. DI is increasing its innovation potential by realizing more efficient neural network designs, Christmann is increasing its capacity through the development of new server designs with significant emphasis on highly configurable heterogeneous server systems, which have potentially lower TCO, and Maxeler is increasing its capacity by supporting more software development models which can exploit its hardware. This may lead to new opportunities for these three partners.	No action to be taken - just a note.
R25	There has been little demonstrated engagement with SMEs and mid-caps outside the consortium and it remains unclear that the project can have broader impact. This is	We will engage more SMEs and mid-caps from different target markets and will report in D6.4.

	acceptable for the initial stages of the project, but as the project evolves, the consortium should try to engage with other SMEs and mid-caps outside the consortium.	
R26	The work carried out in the project supports increased innovation capacity for the partners involved. For the commercial partners, some specifics are noted directly above. For the non-commercial partners increased innovation capacity is visible for HZI, which could potentially increase significantly the biomarker discovery rate, for UniBe, which has a compelling Smart Mirror demonstrator, and for BSC, which can support application development and management for more heterogeneous hardware in a HPC context. The more experimental work of Technion (DFiant) and Chalmers (XITAO) is progressing and may receive validated within the project as good solutions to their respective problems.	The individual exploitation plan will be provided in M20 D6.3 Exploitation Plan
R27	The current progress of the project is in line with the environmental policy objectives and strategies by contributing to the implementation of measures for the reduction of computational power consumption. The results of the project could be well interesting for policy makers dealing with energy efficiency in smart buildings, including office and public buildings. The results from the smart home use cases can be extrapolated to them. The results from the SmarCity use case may also be interesting for policy makers and public authorities regarding pollution management in big cities.	Add possible explotation of the results obtained in the SmartCity use case by other external projects. We will add that CLASS project might get some benefits from using the LEGaTO SmartCity Use case knowledge and results.
R28	The project has not demonstrated clear efforts to achieve gender balance within the action. The reviewers note that it is notoriously difficult to achieve real gender balance within this heavily male-dominated field, particularly in a project, which has a very strong scientific and technical focus. However, the project team could make more effort to improve the male/female ratio within the consortium.	Project will keep working on this issue. Specific comments added in D1.1.

Implementation

R29	the quality of the deliverables produced to date has been inadequate raising questions about the execution of quality processes	The quality revision process will be applied in every deliverable submission as it was initially defined.
R30	milestone planning is too coarse-grained to understand clearly if a significant milestone or achievement has been made	A new project-internal milestone was added
R31	innovation and IPR management requires more precision and should not follow an approach which focuses solely on LEGATO as a holistic solution	Created two repositories in SVN to collect information from the partners regarding IP and software components and will follow up and manage it accordingly
R32	the interaction with the IAB is not clear.	Addressed in R2
R33	Security, Performance and Energy Trade-off of Hardware-assisted Memory Protection Mechanisms. 15th ACM International Conference on Computing Frontiers. ACM. 2018. However, this appears to have been published at an IEEE conference in Brazil.	Not in Open Access. Neuchatel has been informed.
R34	Salami, B., O. S. Unsal, and A. Cristal Kestelman. Comprehensive Evaluation of Supply Voltage Underscaling in FPGA on-chip Memorie. The 51st Annua IIEEE/ACM International Symposium on Microarchitecture (Micro). 2018. This appears to be a lightning talk rather than a classical publication – this should be made clear.	https://legato-project.eu/publication/comprehensive-evaluation-supply-voltage-underscaling-fpga-chip-memories (the conference opened with a lighting talk session so the attendees could decide which talks were more interesting, the detailed talks were scheduled later and we not publicly available)
R35	Colmant, M., R. Rouvoy, M. Kurpicz, A. Sobe, P. Felber, and L. Seinturier. The next 700 CPU power models. Journal of Systems and Software, Volume 144, 2018, Pages 382-396. Elsevier. 2018. This does not look to be strongly linked to LEGaTO and the preprint available from INRIA has no LEGaTO credit.	The publication has been published in the website: https://legato-project.eu/publication/next-700-cpu-power-models
R36	Salami, B., O. S. Unsal, and A. Cristal Kestelman. On the Resilience of RTL NN Accelerators: Fault Characterization and Mitigation. High Performance Machine Learning (HPML) Workshop in conjunction with 30th	The publication has been published in the website: https://legato-project.eu/publication/resilience-rtl-nn-accelerators-fault-characterization-and-mitigation

	International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD). 2018. This has no DOI on the LEGaTO website.	
R37	Dissemination and communication in social media has been performed through the social media channel of the partners and associated entities, but the project lacks its own social media channels. See comments to D6.1 in Annex A.	LinkedIn and Slideshare accounts created. A social media plan will be prepared in the following months.
R38	A dissemination plan has been prepared (D6.1) which in general fulfil the expectations but needs to be updated according to the comments related to Expected Impact 3 in section 3 and the specific comments to D6.1 in Annex A.	Update of the dissemination plan with the target audiences done.
R39	Concerns exist regarding the current IP and market identification that need to be taken into account in the final exploitation plan:	The exploitation plan will be sent in M20, and will include this information.
R40	A Data Management plan has been provided as a deliverable (D6.2); however, it refers quite exclusively to the data sets that will be used in the use cases but it is expected that other data sets could be generated throughout the project. During the review, a data set relating to the undervolting of the FPGA was discussed, which could be made available; also there will be data sets pertaining to OmpSs graphs produced, perhaps data sets relating to application performance analysis etc.	All the data regarding OmpSs benchmarking will be made accessible openly. The D6.2 has been updated accordingly.

Annex I D1.1 (for deliverable rewrite)

R41	The involvement of the IAB seems relevant and appropriate. However a some of points need to be addressed: ● D1.1 specifically states that the IAB is involved in initially phases of the project, providing advice on the prioritisation of these requirements based on industry roadmaps. However if there has been any contribution in that direction it is not visible in D2.1. ● D1.1 does not provide a precise indication of when the IAB contribution are expected (only a generic indication is provided "In the initial phase of the project") nor how the IAB contributions are to be fed back to the documents (produced or being produced).	Addressed in R2.
R42	Despite reasonable presentation of quality processes (§3.2.4), the documentation produced to date has been inadequate from a quality perspective: ● Executives summaries need to be more concise and limited to highlight the essentials of the deliverable ● Conclusions need to clearly summarize the main points and how they are aligned and contribute to the overall objectives and impacts described in the DoA. ● The project needs to ensure future outputs undergo appropriate review.	Addressed in R1 Past Deliverables will be reshaped in this format and the new ones will strictly follow this structure and indications.
R43	Software quality needs to be addressed: ● The project should also consider how to ensure good quality software is produced; note that it is not expected that the output of an R&D project is necessarily production quality software (with 95%+ test coverage), but we do have an expectation that there is some documentation and some test coverage such that it can be used by others.	HZI will provide documentation and testing through validation of the developed software. CHR has already implemented a continuous testing process for its software development.
R44	While there does not exist a real concern on the management of internal communication (§3.1) using email, the consortium members are encouraged to consider the use of collaborative and team communication tools enabling more effective and immediate communications.	A dedicated Slack channel was opened with different sub-channels.

Annex I D6.1 (for deliverable rewrite)

R45	While the target audience is identified (§4), there is not information on which communication channels (§5) are used to address each of these groups, while this is a key element for the effectiveness of this communication.	The dissemination plan includes an exhaustive table with this information
R46	Social networks (LinkedIn, Twitter, SlideShare, Youtube) are very powerful tools, which are not currently exploited in LEGaTO. They should be incorporated in the project to channel communications produced specifically for other dissemination activities (e.g. scientific publications, press clippings, hackathon, workshops, etc.). Basic mechanisms such as pushing out all slide decks to slideshare and publishing them on linkedin require very little effort and can have reasonable impact. EC's report "H2020 Guidance - Social media guide for EU funded R&I projects" provides useful guidelines for the development of a social media strategy.	SlideShare has been created. A social media plan will be prepared in the following months.
R47	It is clear project has a strong OmpSs focus, but the claimed focus is on application developers and the OmpSs community is not large. The project needs to carefully consider how it can maximize its impact regarding the large set of developers, that is it needs to be more specific with respect to which sets of application developers could obtain benefit from the LEGaTO technologies.	We will expand the focus to larger communities through exploiting existing synergies (for example we will target the OpenMP community, leveraging the role of OmpSs as a testing vehicle for extensions to OpenMP standard)
R48	The project uses a reasonable amount of open source software; however this is not at all apparent from the project website - the project should provide a github repo which forks software repos as necessary, provides some overview on how they can be used together (most probably not fully integrated, but some integration is expected)	LEGaTO Github created, all software is linked to it: https://legato-project.eu/software-components
R49	The project needs to identify an appropriate approach to giving sufficient credit to LEGaTO for work contributing to another code base (e.g. make a dedicated fork for LEGaTO which could get periodically merged with the main codebase)	Will be done for LEGaTO contributed code (this will be the case for example for the FTI checkpointing library)
R50	The consortium could also consider the IAB as a potential ally in the dissemination activities, given their position in the market.	A page has been created: https://legato-project.eu/about/industrial-advisory-board IAB member logos will be added soon.

Annex I D6.2 (for deliverable rewrite)

R51	Project needs to be more open with respect to data sets that it will produce: these do not have to be very large data sets, but the result of their R&D activities should be driven by data and hence this data should default to open unless there are some significant commercial sensitivities.	Each generated data will be questioned to be open and if not, it will be justified.
R52	The document should include a summary table (possibly in a conclusion or introduction section) summarising all the data sets that will be used (collected, produced or already available) and for each of them collect the relevant information provided along the DMP. This information should cover at least: data origin (project task/WP or already available) whether the data will be openly available, where it will be available, interoperation formats (if any) and license.	Included in the Executive Summary.
R53	The license for openly available datasets needs to be clarified. Now the document states “GPL- alike” but misses to identify a specific license. “ODC Open Database License (ODbL)” seems to be a good candidate for that.	Comment and license included in the deliverable.

Annex I D2.1 (for deliverable rewrite)

R54	In general terms, the document quality should be improved according to the comments already provided for D1,1: ● The executive summary is too long and does not fully accomplish its purpose (providing an overall idea of the contents of the documents, the benefits of LEGaTO, and where do they come from). ● A proper summary was missing at the end of the document (an updated version was provided on the day before the the review at the request of the reviewers).	(see R1, R42) Edited the D2.1 Executive Summary to make it more crisp and included discussion about achievements during the period.
R55	There needs to be some summary of the applications at the end of this chapter. We suggest a table which includes application name, language(s) application is written in, which legato components will be used by the application, which components are targeted for optimization,	Table 3.4 added.
R56	There is a clear emphasis in LEGaTO regarding power consumption reduction, however the document does not clearly present which is the current power consumption baseline for all the use cases (an approximation would suffice). The “Smart” use case is an example of that. Having a clear baseline is key to drive the development of the LEGaTO technologies and to evaluate their success.	Table with power baselines were added.
R57	While the DoA specifies an objective of 10x reduction in power consumption, is not clear which uses cases will address that. In the case that a specific one is not going to reach it should provide the intended target.	Updated in the chapter.
R58	The two last points could be addressed by incorporating the necessary information to the table suggested in the first point or be gathered in a different one specifically addressing the power consumption topic.	Done.

R59	Additionally, the “Smart Home” mirror should target a more ambitious power consumption around 50W rather than the 100W target that was discussed at the review.	Done.
R60	The respective sections seem to address all the objectives of the LEGaTO project (power consumption, trusted computing base, MTBF and FPGA designed productivity), but they miss to provide a global view on how they contribute to these objectives. We suggest including a table for each section detailing which component of the LEGaTO technologies contribute to each objective.	Added table 7.1 in the SD conclusion chapter.
R61	The above comment can be extended to the techniques that are meant to contribute to the objectives of LEGaTO (e.g., task replication contributes to MTBF, undervolting contributes to power reduction, OmpSs mapping annotations contributes to FPGA designer productivity, etc.). A summary table can be employed in the same spirit as for the components.	Added table 7.1 in the SD conclusion chapter.
R62	The deliverable presents some energy-related concepts and formulas (§2.3) but they do not seem to be referenced anywhere in the document. They may be of good use related to the concerns described below about power consumption baseline and targets in the uses cases.	We have adapted and moved the energy model to the Backend (WP3) subsection since the model is tightly coupled with the task concept.
R63	Some progress was presented during the review session regarding the aggressive undervolting of FPGA (§2.2, last bullet). It would be nice that D2.1 also gathers these preliminary results.	Section on undervolting added (5.7.4).
R64	D2.1 should include a TCO calculation for the server systems (§6.1.5) that does not involve a 32kW draw on the rack (e.g. reduce the per rack power consumption by half and consider twice the number of racks).	adapted TCO calculation in D2.1.