Logo
Placeholder,
see README

Fakultät für Informatik
Technische Universität München

Logo
Placeholder,
see README

# Integration of Mixed-Criticality into Dynamic Task Migration for Real-Time Embedded Systems

**Octavio Ivan Delgadillo Ruiz**

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines

**PhD (check specific degree))**

genehmigten Dissertation.

**Vorsitzender:**
Prof. Dr. Uwe Baumgarten

**Prüfende der Dissertation:**

1. Prof. Dr.-Ing. Vorname Nachname
2. Prof. Dr.-Ing. Vorname Nachname

Die Dissertation wurde am 01.01.2016 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 10.05.2016 angenommen.

# Abstract

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

# Zusammenfassung

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

# Contents

# List of Figures

# List of Tables

# 1 Introduction

Trends in the automotive industry are shaping the future of cars in a way that electronic and computing devices are becoming increasingly important. In fact, the majority of innovations in the automotive sector are related to electronic systems, either in the form of hardware or software [**ey1**]. Developments such as vehicle electrification, autonomous driving and vehicle connectivity are only some examples of automotive applications where computer processing has a big relevance [**pwc1**]. It is therefore reasonable to seek for approaches to ensure their usage is efficient and safety requirements are met.

In modern cars, dozens of computers, commonly known as electronic control units (ECUs), execute the various computing tasks present in a car. This number is likely to increase if we consider past trends: as the tasks performed by ECUs often have safety-critical constraints, such as real-time capabilities, they are commonly integrated to serve a specific purpose, avoiding conflicts caused by the parallel execution of other tasks [**vipin1**; **vipin2**]. However, this strategy has shortcomings in the form of inefficient usage of the electronic devices (many tasks are only executed in specific and rare situations), reduced fault tolerance if a system fails, and increased weight and cost of the vehicle due to the high number of ECUs and cables [**vipin2**; **baunach1**]. Hence, it is an important topic in the automotive industry to find solutions for these issues, especially optimizing costs, while ensuring vehicle safety is kept [**mckinsey1**].

ECU consolidation is an approach for the reduction in the number of electronic devices in the car. The idea is to consolidate the execution of the tasks from many single-purpose ECUs to a few powerful, multi-purpose ECUs. However, the implementation of ECU consolidation raises other challenges: as more tasks need to be executed on the same platform, higher computation and safety requirements must be met. Regarding safety, it is important to consider the added complexity, and cases where a hardware or software failure occur need to be considered, since a single failure could block many tasks and cause major issues. Also, some safety-critical tasks require redundancy to ensure their correct execution [**mundhenk1**]. These challenges have motivated previous research at the chair of operating systems of the Technical University of Munich, where research on projects such as KIA4SM and MaLSAMi has explored the concept of dynamic task migration as a process where the execution of any tasks could move at any given time from one ECU to another. This would allow tasks to finish execution in case an ECU is overloaded or stops working, effectively allowing them to finish executing and meet their real-time constraints.

Previous work at the chair has divided the task migration process in two stages: planning and execution. The planning is the stage which generates a task distribution to the devices that will execute them. The execution is then responsible for allowing tasks to migrate from one device to another, while ensuring their progress is not lost and the execution continues and finishes correctly at the target device. So far, the work performed at the chair has explored the approach with a single criticality level, using different priority strategies for scheduling the tasks on an ECU (that is, deciding which task should get the processor and be executed at a given time). While this idea is in itself an important contribution, it is relevant to note that in many safety-critical embedded applications, such as the aerospace and automotive industries, different criticality levels exist, which are often defined in standards such as the ISO 26262, which defines the ASIL (automotive safety integrity level). For example, in a car, the correct functioning of the ABS is highly critical, as a failure could be fatal. In contrast, the functionality of the radio system has a lower criticality, as an eventual failure would not cause any issue other than an unpleasant trip. This idea is also important because in these industries there exist certification agencies that validate a system before it can be distributed.

For this reason, and due to the relevance of the concept of mixed-criticality in many industries, it would be important to expand the research on task migration to explore its integration. In particular, expanding the execution stage to ensure tasks with higher criticality are always able to execute properly and meet their deadline is crucial. But also it is important to find a balance between resource efficiency and reliability of the migration process for different criticality levels. Therefore, the work proposed as part of this PhD program aims to implement a strategy for the migration execution at different levels of criticality. Additionally, the mixed-criticality concept has to be integrated as well into the planning stage. This will be also explored in the scope of this research, although the main focus will be in the execution stage.

Although the work performed as part of this research is based on previous developments at the chair and will likely extend existent tools, it includes the development of a platform that implements mixed-criticality scheduling, as well as a migration system that involves the concept in different areas in the planning and execution stages. This includes the selection of hardware and software, as well as network interfaces. Also, verification of the algorithms used will be performed to ensure safety and timing requirements are met.

## Related Work

At the chair, previous research projects, such as KIA4SM and MaLSAMi, have explored the possibility of migrating tasks running on a device to another in a real-time capable system (for example, ECUs in a vehicle) under certain conditions. As mentioned before, in these works, the migration was divided into two main stages: The first is the migration planning, which determines the hardware that tasks will be migrated to, should the original hardware not be able to fulfill its duty (for example, if there is a failure in that hardware or if the real-time constraints or deadlines would be violated). The second is the execution of the migration, which ensures that corresponding tasks can be migrated from the source device to the target device while keeping their current state.

The migration execution has been explored previously at the chair, for example in projects KIA4SM and HaCRoM. This was explored in the form of a real-time checkpointrestore mechanism. This involves creating and storing a snapshot of running tasks in a shared memory or copying the memory from a device to another. The work is based on Fiasco.OC and Genode OS.

MaLSAMi and subsequent theses analyzed migration planning, with researchers performing schedulability analysis based on machine learning (specifically, on neural networks). Machine learning was picked over traditional mathematical approaches such as the ones proposed by Buttazzo [**buttazzo1**], because the recurrent calculations can become too complex for complex tasks and for big task sets, and they often lead to a pessimistic calculation of the system utilization. By predicting the feasibility of a task set using machine learning algorithms, potentially faster but less precise results are obtained, as demonstrated by previous theses by Taieb [**taieb1**], Utz [**utz1**] and Blieninger [**blieninger1**]. The predictions provided by the machine-learning approach indicate whether a task set is 100% schedulable or not, but they are not completely safe, since false positive predictions may occur. This approach could be a potentially powerful solution for enabling the execution at run-time of the real-time capable migration planning.

Additionally, a few different platforms and setups have been explored in these projects. The used operating systems running on the ECUs are Genode OS, as used in MaLSAMi and a few theses, a real-time operating system based on an extension of Genode OS with Fiasco.OC, as used in KIA4SM [**kia1**] and HaCRoM, and FreeRTOS, as used by Delgadillo [**delgadillo1**]. These developments are considered in the selection of the platform.

These projects have achieved research-relevant results in their segments, but the concept of mixed-criticality has not been explored in related chair internal work. It is therefore necessary to look at relevant chair-external work. In particular, those regarding mixed-criticality scheduling strategies and related to task migration are reviewed next.

Until last decade, the concept of mixed criticality and its involvement in the scheduling of tasks has been mostly explored for single processor systems. In 2011, Baruah et al. proposed an adaptive mixed-criticality scheduling algorithm which set different WCET for different criticality levels of the same task, changing to a higher criticality mode if a job would not report completion after its low criticality deadline [**baruah2**]. Then, also in 2011, Baruah et al. proposed a mixed-criticality scheduling algorithm called EDF-VD for a single processor, which introduces a high criticality mode that modifies high criticality task deadlines to give them a higher priority over low criticality task [**baruah1**]. This algorithm is defined for any number of criticality levels. In 2012 Li and Baruah proposed an extension of the EDF-VD scheduling algorithm to multiprocessors by adding fpEDF strategy, which adds higher priority to tasks with utilization values higher than 0.5 [**libaruah1**]. Their approach did not explore migration strategies, but was one of the first implementations of a mixed-criticality scheduler on a multiple processor system. In 2015, Gratia et al. adapted RUN, a global multiprocessor scheduling algorithm, to support mixed criticality [**gratia1**]. This approach also considers only a dual criticality system, and only considers the migration of low criticality tasks.

In 2018, Ramanathan and Easwaran investigated an approach for mixed-criticality scheduling on multiple processors that involved partial migration based on fixed partitions for some tasks and leaving some low criticality tasks free to execute on any available resource [**ramanathan1**]. In 2019, Zeng et al. tried a similar approach [**zeng1**]. However, their approaches only considered dual-criticality; alsoonly low criticality tasks would migrate from one processor to another, to ensure these tasks also get execution time, even when processors enter high criticality mode and focus on finishing high criticality jobs. In the case of high criticality tasks, they were mapped statically to a single processor.

It is worth noting that to my knowledge, research on mixed-criticality task migration of high criticality tasks, especially applied to the idea of ECU consolidation, has not been published. A possible reason for the lack of research in this area is the fact that nowadays safety is valued much higher than resource efficiency, thus overseeing the potential for optimization in this aspect. However, in my opinion, it should be possible to achieve both goals by implementing a holistic strategy and therefore it is an area where research can contribute importantly.

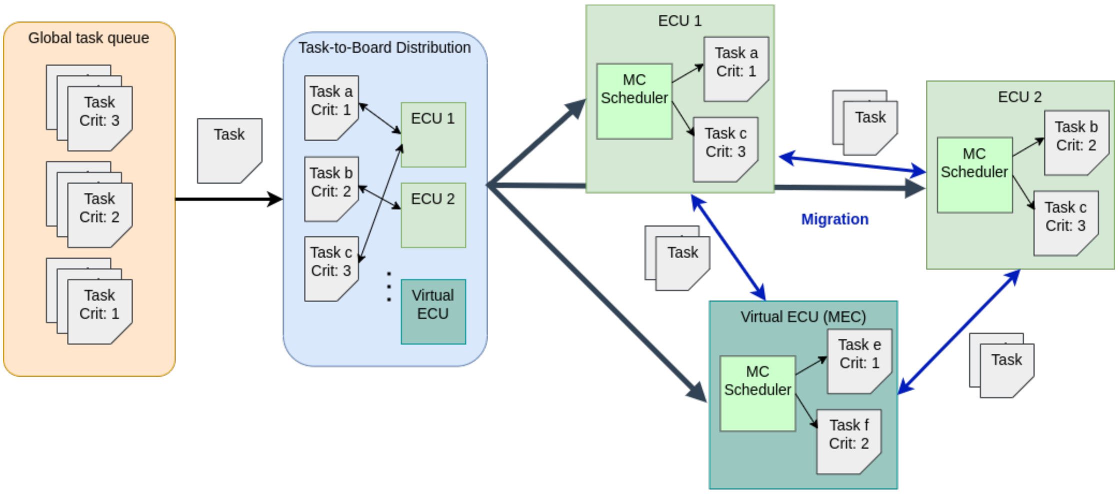
# 2 State of the Art

Example citations [**barham2003xen**; **LIS**]. Example acronym usage Central Processing Unit (CPU).

# 3 Approach

The proposed system is a "breathable" distributed system, where breathable means that it can react at runtime to distinct situations to reconfigure itself and update a decision of where the tasks will execute, always prioritizing the execution of tasks based on their criticality. The situations that trigger this breathable behavior are mainly categorized in two types: 1) changes to the set of tasks that the system must execute (task queue) and 2) changes to the availability of computing resources. In the first category, it is assumed that all tasks have at least soft real-time behavior and they can all be categorized as periodic (need to be executed every certain time and constantly), sporadic (are executed upon request and only once) or a combination of both (might be triggered at a certain point in time and execute for a certain number of times with a period before finishing the processing) [**find a reference to support this assumption**]. The second category includes reacting to addition or removal of physical devices or hardware (for example, due to hardware failure or device restarts triggered by updates, etc.) and also to the availability and latency of virtual devices in the mobile-edge-cloud (for example, when driving over a highway section with MEC support).

These characteristics of the system can be summarized in the image below (add proper reference). The system behavior can be then described as follows. First, a reconfiguring situation triggers a re-distribution of the tasks that need to be executed, which are kept in the global queue. This global task queue as well as the set of both physical and virtual devices (or ECUs) are fed to a task partitioning algorithm, which decides a task-to-board mapping and then triggers the distribution of the tasks on the respective mapped device. The final step is that, depending on the previous mapping of tasks to boards, if the assigned ECU for a task changes, it is necessary to migrate its execution from the previous device (source) to the new one (target). This last step is the main objective of this research work, and in the scope of it, a few requirements should be considered to deem the implementation as successful: 1) the task has to be stopped in the source device and started in the target device, 2) the relevant task context should be also migrated, meaning that no crucial data is lost (in the case of a SLAM algorithm, the map and position would not be lost), 3) both the migration decision and execution strategies should consider the criticality of the tasks, 4) real-time behavior should be also considered in the migration strategy. It is relevant to mention that a master ECU would be responsible for both keeping the global task queue, triggering the task-to-board distribution algorithm (which itself can run on a separate ECU or a specialized ML hardware) and keeping the relevant files to start tasks when necessary.



Based on the previous list of requirements, and especially 3 and 4, at least 2 migration strategies should be implemented, since tasks with different criticalities also have different constraints to be prioritized. For example, in the case of low-criticality tasks, it might be more relevant that the task does not end in starvation, whereas for a high critical task it is crucial that the result is delivered always on-time and that, if the task needs to be paused due to a critical failure, it is brought back up in a reduced and deterministic time to keep up with the real-time constraints.

## 3.1 Overview of the Approach

As mentioned before, the main objective of the proposed research is to achieve the integration of mixed criticality into the task migration process. The proper integration would require different strategies for the different criticality modes. For this reason, it is first important to define what is understood under mixed criticality in the scope of this work. While many publications consider only two criticality levels (high and low), the standards commonly describe several of them. In this work, the concept of multiple criticality levels will be used in a conceptual phase following the ASIL standard in ISO 26262, while the implementation and results will be collected for at least 2 levels in both the scheduling algorithm and the migration strategy. Whereas a model with multiple criticality levels represents real situations better over one with just two, as tasks can have intermediate criticality levels, in the scope of this work, the focus relies more on fulfilling challenges at both ends of the criticality spectrum, and assuming a mix of both strategies might apply as a solution for intermediate levels. One concept that should be covered is the consideration of different execution times according to a criticality level, as normally higher criticality levels offer pessimistic WCET estimates to ensure correct behavior occurs, but actual execution times are often lower. This is a concept that is explored in many publications and is often necessary to ensure compliance with certification authorities.

The first element needed for the system developed is the integration of a mixed-criticality scheduler for each of the physical devices. This should ensure that for any given device, the tasks with a higher criticality will never miss the deadline for a job, while being more flexible for lower criticality levels. In this regard, exploring different mixed-criticality scheduling algorithms and their integration in the development platform is an important step that will allow the migration to occur and be evaluated properly. Several publications have proposed mixed-criticality scheduling algorithms, such as [**baruah1**; **fleming1**; **zhao1**; **baruah2**; **lili1**], and a few of them can be implemented and compared. It is also important to mention that following the state of the art in embedded devices, a multi-core implementation is desired, with asymmetrical multiprocessing (AMP) being the preferred solution, as software failures could be isolated to affect less tasks (***find a reference***). In particular, following approaches are explored (so far): hierarchical virtual deadline EDF (faulty implementation); reserved cores for 2-level MC, with LO crit cores as backup cores for HI tasks. It is assumed that this step should not affect the behavior of the virtual devices, since by concept, they can only run non highly (not necessarily only LO) critical tasks, due to the uncertainties introduced by the external communication with the MEC (explain further...).

Another important element for the distributed system to ensure its compliance to the real-time constraints is a timing protocol such as PTP, to keep the different devices synchronized and share the same notion of time when following task deadlines. This way, even when migrating a task, information on the deadlines should be very accurate. In addition to this, migration strategies must consider that transmission times can be variable and, in the ideal case, the source device should keep the original task execution until it is ensured that the target device has started the execution successfully, potentially taking longer than the deadline of a running job.

The migration planning will be adapted from previous work, namely the student theses supervised by Bernhard Blieninger. As these don't consider mixed-criticality or a multi-core implementation in the system behavior, a few changes are necessary to integrate both concepts. First, the schedulability analysis that is commonly used for selecting the best task distribution has to be extended to work with criticality levels and the use of a mixed-criticality scheduler. Here, it has to be taken into consideration whether extending and retraining the machine learning models is worth the effort, especially since this idea is still not proven as a solution and the extension with the criticality levels and other relevant information might add complexity to the net. Otherwise, implementing a mathematical schedulability analysis that considers mixed criticality is necessary. A second extension that should be performed in the planning stage is to penalize the migration of higher criticality tasks, as the migration overhead adds an additional risk for the tasks failing. Furthermore, the availability of both virtual and physical devices needs to be considered at the moment of deciding which tasks should migrate and where.

The next step in the development of this strategy is the execution of the migration, which is the core concept explored in this work. It is important to consider the necessity to meet strict timing constraints for tasks at the higher end of the criticality spectrum. ***THIS HI criticality solution is yet to be explored, Yinbo is working on the variant for LO crit*** While the exact approach is yet to be proven as feasible, an idea for higher criticality tasks would be to let them run in a standby state in all or a subset of the total of available ECUs, and only executing it in one ECU, while storing runtime generated data either in a central unit or in a shared memory only accessible to highly critical tasks, in this way only a start/stop signal is sent to the task and the execution can be resumed quickly, also ensuring fail-safety in case the executing device fails. Also, for tasks that require redundancy (e.g. ASIL level D), it should be possible for tasks to execute actively multiple times and produce results in parallel. Another addition in this stage should be the introduction of a real-time capable communication protocol. In this sense, it should be possible to bound the migration time for this tasks, ideally in the range of a few milliseconds. These ideas could be faster to execute and allow for less variance in the time, eventually making it possible to perform formal verification, but it would be expensive if tasks at all criticality levels were to run like that, as the resources are utilized inefficiently. This should be acceptable for highly critical tasks, though, since the majority of the tasks would be assigned low to medium criticality levels.

The strategy for migrating lower criticality tasks between physical devices is prone to more flexibility, so that a wider range of tasks can be deployed with a lower impact in terms of resource efficiency, even if this would open the possibility for more erroneous behavior in the tasks. The approach currently explored for tasks with low criticality involves the transmission of precompiled task binaries and a snapshot of the execution data every time a task is distributed to an ECU and the usage of a normal TCP/IP protocol over Ethernet. ***This is Yinbo's ELF loader thesis*** With this strategy, the resource utilization in the devices is made more efficient, but the time spent for the migration is increased as the communication is slower due to the amount of information exchanged, and also less reliable due to the nature of the communication protocol.

As an optional step, the exploration of strategies for more criticality levels is yet to be done, but a few ideas are suggested. First, a combination of the two strategies mentioned could be implemented for intermediate criticality levels, such as leaving the tasks running but using a less predictable communication protocol. Another idea is to subdivide each of the criticality levels mentioned and there perform variations of the mentioned strategy. For example, in a system with a certain number of devices and 2 high criticality sublevels, the highest sublevel tasks are kept in standby on all devices, while the rest are only kept in standby in a few of the devices, making sure there is always an ECU ready for the highest criticality and using less resources for the second sublevel. In the case of lower criticality, this could be implemented in the form of giving priority to the migration and transmission of data of tasks at the higher sublevels.

Additionally, due to the consideration of virtual devices in the MEC, an additional strategy is also implemented in the scope of this work. That is, the migration of low criticality tasks from and to the MEC. For this implementation, it is assumed that the MEC servers share some aspects of the architecture with the real hardware and that the devices can be modeled in the form of containers as their digital twins. In this way, upon availability, the master ECU can request to start a container which will run the required task. It is assumed that this behavior is allowed for low criticality tasks that can benefit from the more powerful resources available in the MEC, and that communication latency can be neglected as long as the results produced are relevant. An example of this might be a path planning algorithm for an autonomous vehicle, where the immediate safety of the vehicle is not threatened, but the algorithm can be extended with further sensor data and better hardware for performing ML tasks. ***Part of this has already been implemented by Roberto*** It is relevant in the scope of the migration, that the virtual tasks have an equivalent task implemented for the real devices, and that the task data can also be migrated between them. In this way, the task can be moved depending on the need and availability of the resources to and from the MEC.

+ ’

# 4 Content

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

## 4.1 Example Section

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris. Nunc sed pede. Praesent vitae lectus. Praesent neque justo, vehicula eget, interdum id, facilisis et, nibh. Phasellus at purus et libero lacinia dictum. Fusce aliquet. Nulla eu ante placerat leo semper dictum. Mauris metus. Curabitur lobortis. Curabitur sollicitudin hendrerit nunc. Donec ultrices lacus id ipsum.

### 4.1.1 Example Table

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

**Table 4.1:** Example Table.

| | |
|---:|:---|
| 7C0 | hexadecimal |
| 3700 | octal |
| 11111000000 | binary |
| 1984 | decimal |

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

### 4.1.2 Example Plot

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

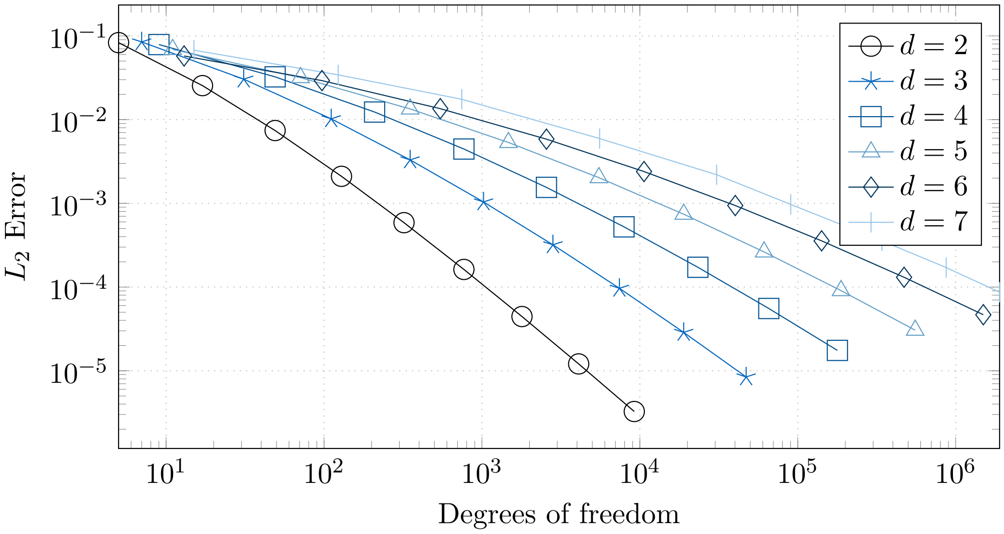


**Figure 4.1:** Example Plot.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa. Nunc sed pede. Praesent vitae lectus. Praesent neque justo, vehicula eget, interdum id, facilisis et, nibh. Phasellus at purus et libero lacinia dictum. Fusce aliquet. Nulla eu ante placerat leo semper dictum. Mauris metus. Curabitur lobortis. Curabitur sollicitudin hendrerit nunc. Donec ultrices lacus id ipsum.

### 4.1.3 Example Barchart

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

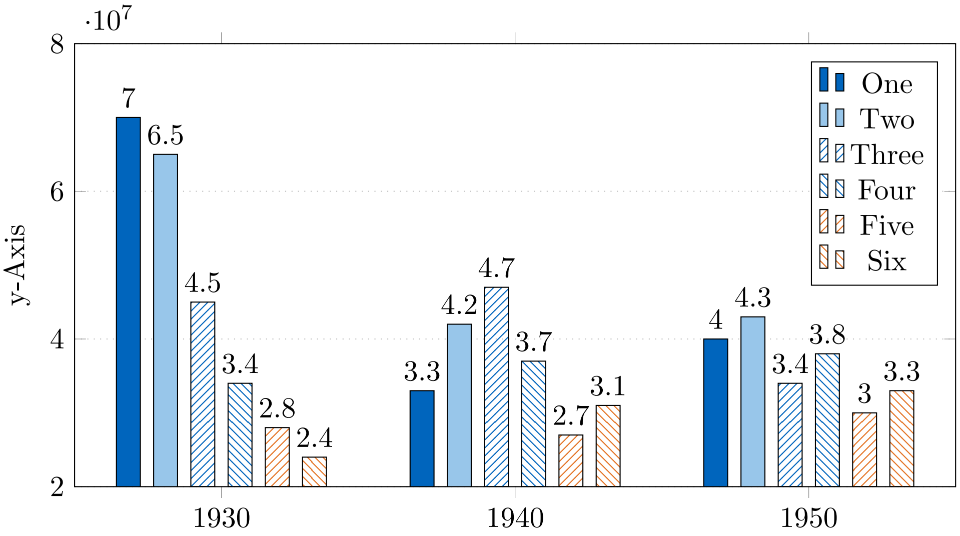


**Figure 4.2:** Example Barchart.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa. Nunc sed pede. Praesent vitae lectus. Praesent neque justo, vehicula eget, interdum id, facilisis et, nibh. Phasellus at purus et libero lacinia dictum. Fusce aliquet. Nulla eu ante placerat leo semper dictum. Mauris metus. Curabitur lobortis. Curabitur sollicitudin hendrerit nunc. Donec ultrices lacus id ipsum.

### 4.1.4 Example SVG Graphics Input

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

**Figure 4.3:** Example Shapes from an SVG.

Quisque consectetuer. In suscipit mauris a dolor pellentesque consectetuer. Mauris convallis neque non erat. In lacinia. Pellentesque leo eros, sagittis quis, fermentum quis, tincidunt ut, sapien. Maecenas sem. Curabitur eros odio, interdum eu, feugiat eu, porta ac, nisl. Curabitur nunc. Etiam fermentum convallis velit. Pellentesque laoreet lacus. Quisque sed elit. Nam quis tellus. Aliquam tellus arcu, adipiscing non, tincidunt eleifend, adipiscing quis, augue. Vivamus elementum placerat enim. Suspendisse ut tortor. Integer faucibus adipiscing felis. Aenean consectetuer mattis lectus. Morbi malesuada faucibus dolor. Nam lacus. Etiam arcu libero, malesuada vitae, aliquam vitae, blandit tristique, nisl.

### 4.1.5 Example Source Code

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

```c
#include <stdio.h>

void quick_sort (int *a, int n) {
    int i, j, p, t;
    if (n < 2)
        return;
    p = a[n / 2];
    for (i = 0, j = n - 1;; i++, j--) {
        while (a[i] < p)
            i++;
        while (p < a[j])
            j--;
        if (i >= j)
            break;
        t = a[i];
        a[i] = a[j];
        a[j] = t;
    }
    quick_sort(a, i);
    quick_sort(a + i, n - i);
}

/* main routine */
int main (void) {
    int a[] = {4, 65, 2, -31, 0, 99, 2, 83, 782, 1};
    int n = sizeof a / sizeof a[0];
    int i;
    for (i = 0; i < n; i++)
        printf("%d%s", a[i], i == n - 1 ? "\n" : "␣");
    quick_sort(a, n);
    for (i = 0; i < n; i++)
        printf("%d%s", a[i], i == n - 1 ? "\n" : "␣");
    return 0;
}
```

**Listing 4.1:** Example Source Code

### 4.1.6 Example Subsection

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

#### 4.1.6.1 Example SubSubSection

Morbi luctus, wisi viverra faucibus pretium, nibh est placerat odio, nec commodo wisi enim eget quam. Quisque libero justo, consectetuer a, feugiat vitae, porttitor eu, libero. Suspendisse sed mauris vitae elit sollicitudin malesuada. Maecenas ultricies eros sit amet ante. Ut venenatis velit. Maecenas sed mi eget dui varius euismod. Phasellus aliquet volutpat odio. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Pellentesque sit amet pede ac sem eleifend consectetuer. Nullam elementum, urna vel imperdiet sodales, elit ipsum pharetra ligula, ac pretium ante justo a nulla. Curabitur tristique arcu eu metus. Vestibulum lectus. Proin mauris. Proin eu nunc eu urna hendrerit faucibus. Aliquam auctor, pede consequat laoreet varius, eros tellus scelerisque quam, pellentesque hendrerit ipsum dolor sed augue. Nulla nec lacus.

#### 4.1.6.2 Example SubSubSection

Suspendisse vitae elit. Aliquam arcu neque, ornare in, ullamcorper quis, commodo eu, libero. Fusce sagittis erat at erat tristique mollis. Maecenas sapien libero, molestie et, lobortis in, sodales eget, dui. Morbi ultrices rutrum lorem. Nam elementum ullamcorper leo. Morbi dui. Aliquam sagittis. Nunc placerat. Pellentesque tristique sodales est. Maecenas imperdiet lacinia velit. Cras non urna. Morbi eros pede, suscipit ac, varius vel, egestas non, eros. Praesent malesuada, diam id pretium elementum, eros sem dictum tortor, vel consectetuer odio sem sed wisi.

# 5 Conclusion & Outlook

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

# A Additional Stuff

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.