



Computer Science  
UNIVERSITY OF TORONTO

TIZEN OS SUPPORT  
FOR SOTI MOBICONTROL  
INTEROPERABILITY

***Qi Zhao***  
*Master of Science in Applied Computing*  
*Department of Computer Science*  
*University of Toronto*

# Contents

- The name and short description of the company ..... 1
- Your faculty and industrial supervisors with their contact information ..... 1
  - Academic Supervisor ..... 1
  - Industrial Supervisor ..... 1
- An abstract of about 300 words describing your role with the company using nontechnical English ..... 1
- Your original role and expected outcomes ..... 2
  - Original Role..... 2
  - Expected Outcomes ..... 2
- How your role and expectations changed over time..... 3
- The outcomes you were able to achieve ..... 4
- The research you either incorporated from other sources or performed yourself ..... 4
- The impact of your work on the company..... 5
- Lessons learned from your internship experience, and an overall reflection on the value of the internship to your professional growth ..... 6
- Your next steps. What are you going to do next in your career? ..... 7
- Reference ..... 8

# Tizen OS Support for SOTI MobiControl Interoperability

- **The name and short description of the company**

I'm working for SOTI Inc.

SOTI is a proven leader at creating innovative solutions that reduce the cost and complexity of business-critical mobility and the IoT. Thousands of companies around the world depend on SOTI's product to secure, manage and support their mobile operations. SOTI has built strong partnerships with leading mobile platform providers and device manufacturers. Cooperating with these relationships, SOTI helps businesses take mobility to endless possibilities.

SOTI's main product is SOTI MobiControl. It is an enterprise mobility management (EMM) solution that secures and manages Google Android™, Apple® iOS, Linux®, macOS® and Microsoft Windows® devices, throughout their entire lifecycle, from deployment to retirement. It removes the complexity from managing a multi-OS, multi-vendor and multi-purpose business mobility program.

- **Your faculty and industrial supervisors with their contact information**

Academic Supervisor

- Name: Eyal de Lara
- Email: [delara@cs.toronto.edu](mailto:delara@cs.toronto.edu)
- Phone: (416) 946-8656

Industrial Supervisor

- Name: Lorne Wilkinson
- Email: [lorne.wilkinson@soti.net](mailto:lorne.wilkinson@soti.net)
- Phone: (905) 624-9828

- **An abstract of about 300 words describing your role with the company using nontechnical English**

Business mobility is complicated and becoming more so. Within just a few years, we can expect to see billions of new, connected devices, 'things' and IoT endpoints. SOTI MobiControl delivers management and security for these new mobile devices, sensors and endpoints, as well as their applications and back-end systems. However, more devices mean more kinds of operating systems and back-end platforms, which is a huge challenge for SOTI.

My role with SOTI for this internship project is to expand SOTI MobiControl, by adding support for the Tizen Operating System. Tizen™ is an open-source operating system that

is widely used by Samsung IoT devices because of its lighter weight, faster speed and higher security.

In this project, I have tried to puzzle out the most effective approach to port the existing MobiControl agent from other platforms to Tizen platform. During the first several weeks, with the help of Samsung IT consultants and official documentations, I learned the Tizen system and security model design ideas, as well as the necessary components of establishing a Tizen application. After that, I did a survey of the existing SOTI products, which helped me understand the architecture of existing agents on other platforms. After investigation and comparison, we decided to port based on the Linux agent. However, Tizen is designed for strict security protection. To deal with this, I had to “hack” into Tizen file system, dig around and find out where our agent was blocked. Once our agent can successfully run on Tizen devices, I started to port different functions, such as disabling the use of applications, changing the clock time, sending notifications and so on. Finally, I did functional and non-functional testing, found bugs and fixed them. The goal is to make our new Tizen agent stable enough to be released by SOTI.

- **Your original role and expected outcomes**

#### Original Role

- Set up working environment; get familiar with SOTI MobiControl console and agents from different platforms. Learn to use common features and functions of MobiControl.
- Review and analyze the historical info about porting from one platform to another that has occurred previously in Development. Research on Tizen system, get deeper understanding on its architecture and security model.
- Investigate and develop design studies to address top down and bottom up requirements of porting one of the existing agents to a new Tizen agent.
- Based on the design studies, discuss and brainstorm to find out the best way of implementing the porting process.
- After finishing the main process of porting, perform functional and non-functional testing of the new agent, measuring against baseline data to understand the project success.
- If time permits, investigate the memory, CPU usage and battery life consumption related to the agent. Make the agent run in a stable condition but consume the least resources.

#### Expected Outcomes

- Develop an agent that runs on Tizen system and be able to talk to MobiControl server.

- Port the most useful features and functions from the existing agent to the Tizen agent, for example, sending notifications, disabling device settings, changing date and time, etc.
  - Achieve inbound and outbound remote control between Tizen devices and MobiControl server.
  - Be able to enroll various Tizen devices into MobiControl database.
- **How your role and expectations changed over time**

This project of supporting Tizen OS for SOTI MobiControl Interoperability is new and thus not fully defined, so there happen to be changes in the project from time to time.

At the initial stage, we knew almost nothing about Tizen, so we read related papers and materials from Samsung and other online resources. During the investigation, I found that Samsung has created their own IDE for developers to create Tizen native and Web applications. The IDE is called Tizen Studio and it also consists emulator, toolchain, sample code and documentation. I thought our new agent could possibly be a Tizen native application, so I started to get familiar with Tizen Studio, create applications, sign them with certain certifications and run them on the Tizen emulators. This hand-on experience really helped me a lot when later dealing with various issues of building and deploying our agent on target wearable devices.

The other role I played during project initial stage was to define the scope of our project. Tizen operating system is widely used by Samsung for their different devices, such as the Z series cell phone, the Galaxy series watch, the Samsung smart TV and even the Samsung smart refrigerator. Although all these devices are running Tizen OS, there are still lots of differences between phones, watches, TVs and other IoT devices, as well as various Tizen versions. To make it clear, I created a spreadsheet for our team to compare different Tizen platforms and versions. Combined with customer's demand and possibility analysis, we decided to focus on Tizen wearable devices as our primary target.

After initial research, we encountered the problem of being blocked by Tizen security model. Tizen is a light-weighted operating system, but at the same time, it's also designed to be very security concerned. Unless Linux, we are not allowed to be a "root" in Tizen system. In other words, there are many things we can't do because we don't have supervisor permissions. For example, file system is like a black box to us in Tizen system, and many Linux commands are not supported either. The security sandbox blocked us for several weeks, which was very frustrating. My manager and I had to spend a lot of time on debugging the compiling error and they are mainly created by Tizen security model. In addition, Tizen Studio has not provided developers with any mature debugging tool, which made it even harder to fix bugs. We had to use standard printing method to print out debug log wherever needed, and I wrote some code snippets to catch "stack backtrace signal" from Tizen system calls as our own debugging tool. This process was not expected and cost longer than we planned, although it was painful, we made it successfully at the end.

Design studies are made along with the process of porting. It's not a pre-defined document that our team should follow exactly. Due to the workload and porting progress, we were constantly revising the document. For example, while I was about to achieve a feature in Tizen wearable agent, Samsung announced a new version of Tizen SDK for wearables and many interfaces were influenced by the update. Therefore, I had to switch my role to analyze the latest SDK and all the interfaces, trying to figure out which part of our code needed to be changed accordingly and what new features we could possibly implement in the future.

Right now, we are at the testing process. The functional and non-functional testing are still on the way, and there are more issues to be fixed as well. My current role is to continue focusing on testing, creating more bugs and fix them. There might not be enough time for me to do analysis on the resource consumption of the agent.

Also, owing to the changes of Tizen SDK, we are not able to implement auto-restart and auto-reboot features of our Tizen agent. This means once the agent is dead for some reasons, it won't restore by itself. However, as a device management agent, it should always be alive. We will need to cooperate with Samsung to make this fixed.

- **The outcomes you were able to achieve**

We have created our "Alpha" version of the Tizen wearable agent ported from original Linux agent.

The agent can perform many functions. Most of them are ported directly from Linux agent such as rebooting the device, showing applications installed on the device, monitoring device system info, blacklisting applications, installing and uninstalling applications, as well as the most important remote control session. The rest of are specially designed for wearable devices, such as sending notifications, changing watch faces, responding to real-time sensor data, etc.

We have deployed our Tizen wearable agents to Tizen emulators and Samsung watches. All features perform well on the real watches, and all watch models are supported (Samsung has Galaxy Gear S3, Galaxy Watch, Galaxy Watch Active 1 and 2 in the marketplace), but some functions are restricted on emulators because of their different architectures.

We have also started to investigate on the possibility of porting Tizen TV agent. More detail will be delivered to me next year when I become a full-time employee.

- **The research you either incorporated from other sources or performed yourself**

The project starts with the investigation on Tizen operating system. Tizen is a lightweight operating system<sup>[1]</sup>, it's open and flexible, and it's built from the ground up to address the needs of all stakeholders of the mobile and connected device ecosystem<sup>[2]</sup>. Tizen has a strict security model, composed with three different parts: Cynara, Smack and DAC<sup>[1][3]</sup>.

The security model makes sure the following criteria: all the 3rd-party apps are required to obtain the author and distributor signatures; apps are only available on Tizen Store; for testing purpose sideloads, developers must create the keys with the device ID of the testing machine<sup>[4]</sup>. Tizen system is also facing the hardware constraints because it usually runs on mobile or IoT devices where there is no access to sufficient power, memory and storage space.

Another part of learning is about SOTI MobiControl. If we want to port it to other platforms, we should first know how it works now and what features can be achieved. I learned to use MobiControl directly from SOTI's internal resources. I was invited to a bootcamp of SOTI MobiControl where I was trained to setup, deploy and use it for several days. A brief introduction of SOTI MobiControl can be found on our website<sup>[5]</sup>.

The focus of my project is porting. I did research on how to port a software from one platform to another. Of course, there is no specific and standard process of how to port. However, for the company's future convenience, I developed a repeatable porting process according to our own situation. It may not be perfect, and it's not mature, but it did help our project a lot. The process starts with a decision of whether to port the current MobiControl agent or to redevelop a brand new agent. We need to analyze the possibility of both choices. Porting can save lots of time and resources, but different developing environments like compiling and installing tools, different user interfaces, as well as different physical limitations are all the hurdles needed to be overcome<sup>[6]</sup>. Secondly, after deciding to port, we should choose a baseline platform of porting. SOTI has developed MobiControl agents cross various platforms including Android, iOS, Windows, macOS and Linux. Each platform has its own attributes, after comparison among all, plus that Tizen is derived from Linux, we chose Linux as the baseline platform. However, since the user interface and application management system of Tizen watches are similar to Android phones, Android agent sometimes would also be our source of reference. Thirdly, since Tizen watches have lots of limitations like limit battery life, smaller RAM and memory, we should try best to make porting code clean and smart. There should be more run-time dynamic compiling instead of static compiling, and less redundant code snippet to save resources. After successfully generating the agent, new test methods also need to be performed. Unless Linux, it's better to use an Android-like method with more restrictions to do functional and non-functional testing to the new Tizen agent. Last but not least, documentation is important, too. One difficulty for our project is that we have limited reference to learn from. Thanks to the success of this porting process, we can provide documentations to future developers who need to do porting on other platforms. Also, during the process, we always remembered to make our code portable<sup>[7][8]</sup>, which will also benefit the possible future porting work.

- **The impact of your work on the company**

SOTI has just launched this project when I joined 8 months ago. With no previous experience but only very limited materials, we started from scratch. We have not only proved this idea can be achieved but also made it run on real devices successfully.

Thanks to the success of this project, SOTI has successfully brought its Star Product SOTI MobiControl into IoT world. This is a big step for SOTI, not only because there is one more supported platform of MobiControl, but also opens a broader market and creates more potential opportunities in the Mobile Device Management (MDM) field. We have to admit that IoT is the next population in mobile market. IoT means more devices and much harder to manage them. If we can seize this opportunity in time, with the help of good public reputation and public population of MobiControl, this will be a huge success for SOTI.

From October 7<sup>th</sup> to 10<sup>th</sup>, SOTI held its annual partner and user conference SOTI Sync in Toronto. During the event, the outcome of my internship project ---- our new Tizen wearable MobiControl agent was showed up. Our Tizen Booth was crowded with people who wanted to know how we were able to implement so many things in such little watches within just several months. Attendees from our main customer American Airlines was also very surprised to see all the functions and features that can be performed by our agent. Now that if every pilot from American Airlines wears a watch with our agent deployed, the control center can send pilots real-time notifications about any unexpected situations without distracting them, as well as receive pilots' personal health data alert preventing them from unnoticed emergencies.

On November 27<sup>th</sup>, as one of the projects presenting on the Applied Research in Action (ARIA) innovation showcase held by the Department of Computer Science of University of Toronto, our demo drew attention to many attendees. The President and CEO of SOTI, Carl Rodrigues, was so proud to introduce everyone our accomplishment, and called our project "the greatest and most interesting demo in this event". He was satisfied with the performance of our interns and planned to post more positions to the next cohort of MScAC program.

I'm so happy to see our project being recognized by so many people, especially my managers and SOTI's private customers. I'm also glad that SOTI is willing to strengthen the cooperation with University of Toronto and give more opportunities to our next cohort because our accomplishment.

- **Lessons learned from your internship experience, and an overall reflection on the value of the internship to your professional growth**

I'm very glad and excited that SOTI gives me the opportunity to work here as part of a new project.

The first thing I learned is starting small. Facing this big and new project, at first, our team even didn't know how to start. Therefore, I decided to use the only tool we can find about Tizen --- Tizen Studio to start. From the simplest "Hello World" application, to a system info view application, then to a functional application, and finally to an agent application that consists of different functions. Every time our agent needs a new function, I would always start from a simple testing application. Step by step, although each step is small, together they can be a huge jump.



Another important thing is teamwork. We have three people working together for this Tizen porting project, and another three people working on different projects. Every day we would have stand-up meetings to catch up updates on each project. Although we might not work on the same project, I can still always learn from others' ideas. When I had trouble tracing MobiControl logs, the senior MobiControl developer would guide me; when I needed to port some functions from Android agent, our senior Android developer would help me understand Java code base of the agent; when I tried to apply design patterns learned from school, our manager would also give me advice based on his 30-year object-oriented programming experience. I learned from my colleagues every single day, and they helped me become a better developer.

During the developing and researching process, I realized some differences between study and work. The most important one is: don't feel released unless your product is mature and stable enough. For school assignments, all we need to do is to get the expected result and hand it over to the professor. However, for a market product, your job is not just to implement functions, the more important thing is to make sure it's stable and safe. Your product will be used by thousands of clients, all their actions and expectations are unpredictable, so all kinds of testing (including functional testing, non-functional testing, unit test, Ad-hoc testing, etc.) should be considered and well performed.

Last but not least, your manager is important. Working with an experienced, caring and supportive manager would really help you on your job. A good manager would lead you and always give you wise advice along with you career. I'm so lucky that I met one such kind of manager, we had a great time working with each other and I learned a lot from him. He would praise me from time to time, which strongly encouraged me. He would also point out my weakness and lead me a way to catch up. Thanks to my manger, it was a pleasure working with him.

- **Your next steps. What are you going to do next in your career?**

I'm planning to stay in the same team at SOTI.

The internship project is just a starting point for this whole Tizen related projects. We will continue developing more features and functions to the Tizen wearable agent, making it more stable and be able to auto-restart when it fails. We will create an enrollment program for it so that each device can be easily enrolled into our SOTI MobiControl database. Besides Tizen wearable agent, we have also planned to port into Tizen TV agent or other Tizen IoT agents. We will work closely with our partners and customers to deliver the most up-to-date agent.

I'm very glad and honor to be one part of this project, starting from scratch, it's like a new-born baby to us. I will grow together with our "baby" and we will both become better in the future.

- **Reference**

- [1] Ajin Abraham. Hacking tizen: The os of everything. 2015.
- [2] About Tizen. <https://www.tizen.org/about>, 2012.
- [3] Bumjin Im and Ryan Ware. Tizen security overview, 2012.
- [4] Samsung Security Team. Tizen security white paper, 2019.
- [5] SOTI Inc. About SOTI MobiControl. <https://www.soti.net/products/soti-mobicontrol/>, 2019.
- [6] Scott Gartner. What is 'porting' in software development, 2017.
- [7] James D Mooney. Bringing portability to the software process. Dept. of Statistics and Comp. Sci., West Virginia Univ., Morgantown WV, 1997.
- [8] James D Mooney. Developing portable software. In Information Technology, pages 55-84. Springer,2004.