Table of contents

List of figures

General Introduction

Whether we are online shopping, streaming videos or reading the feed on social media, every internet activity involves huge amounts of data that needs to be stored and processed somewhere. Datacenters are there all over the word to accomplish this mission. They have already spread from virtually nothing 10 years ago to consuming about 3 per cent of the global electricity supply and accounting for about 2 per cent of total greenhouse gas emissions. That gives it the same carbon footprint as the airline industry[1].

" *If we carry on going the way we have been it would become unsustainable – this level of data centre growth is not sustainable beyond the next 10 to 15 years. The question is, what are we going to do about it?* "

Professor Ian Bitterlin [1]

We have already improved everything around compute architecture, the only thing left to improve is the compute side of the equation. It's either a breakthrough in our perspective about it, or we need to get deadly serious about doubling the number of power plants on the planet. One way to curb their carbon footprint is to increase the amount of renewable energy they use. But even if the industry was able to shift to 100 per cent renewable electricity, the volume of energy they would need would put intolerable pressure on the world's power systems. So which way the industry decides to go could have a huge bearing on whether renewable energy receives the huge investment that drives innovation – bringing down the cost of green electricity to everybody's benefit. It could also play a large role in determining whether the world can avoid the worst ravages of global warming.

[1] Professor Ian Bitterlin is a Chartered Engineer with more than 25 years' experience in data-center power and cooling, CTO of Emerson Network Power Systems in EMEA and a Visiting Professor at University of Leeds in the School of Mechanical Engineering.

In this context, this project, originally suggested by iExec, is performed as part of the preparation to obtain the computer science engineering degree from the Faculty of Mathematical, Physical and Natural Sciences of Tunis, University of Tunis ELMANAR, Tunisia. It aims to suggest a solution to this problem in an innovative approach by embracing Edge computing concept combined with solar energy.

This report illustrates the work that has been done, from design to requirements and implementation. Those topics are covered in five chapters, we start by contextualizing the project and discussing the state of the art with a comparison between our perspective and some existing ones. Then, we present the different requirements and technical specifications which leads us to the implementation details. Finally we conclude and suggest some improvements to the current limitations.

Chapter 1

Project Context

1.1 Introduction

The aim of this chapter is to contextualize the project by giving its background, to present the host company and to define in a later section the main problem as well as the way we adresse it.

1.2 Project background

This project is about desining and implementing a positive energy worker. A Raspberry Pi based system that powers iExec's infrustructure and serves - at the same time - as an IoT device to embraces the Edge computing concept. It is achieved in the context of the preparation of the end of studies project submitted to obtain computer science engineering degree.

1.3 Host company

Fig. 1.1 iExec logo

Started in 2016, iExec[2] was co-founded by Dr. Gilles Fedak and Pr. Haiwu He. Ph.D, CEO and Co-Founder, Dr. Gilles Fedak has been a permanent INRIA research scientist since 2004 at the ENS in Lyon, France. His research interests lie in Parallel and Distributed Computing, with a particular emphasis on the problematic of using large and loosely-coupled distributed computing infrastructures to support highly demanding computational and data-intensive science. He co-authored about 80 peer-reviewed scientific papers and won two Best Paper awards. Pr. Haiwu He, Ph.D, Co-Founder and Head of Asian-Pacific Region was a research engineer expert at INRIA Rhone-Alpesin Lyon, France from 2008 to 2014. He has published about 30 refereed journal and conference papers. His research interest covers peer-to-peer distributed systems, cloud computing, and big data.

iExec aims at providing decentralized applications running on the blockchain a scalable, secure and easy access to the services, data-sets and computing resources they need. This technology relies on Ethereum smart contracts and allows the building of a virtual cloud infrastructure that provides high-performance computing services on demand.

iExec leverages a set of research technologies that have been developed at the INRIA and CNRS research institutes in the field of Desktop Grid computing. The idea of Desktop Grid (aka. Volunteer Computing) is to collect the computer resources that are underutilized on the Internet to execute very large parallel applications at the fraction of the cost of a traditional supercomputer. iExec relies on XtremWeb-HEP[3], a mature, solid, and open-source Desktop Grid software which implements all the needed features: fault-tolerance, multi-applications, multi-users, hybrid public/private infrastructure, deployment of virtual images, data management, security and accountability, and many more.

iExec is developing a new Proof-of-Contribution[4] (PoCo) protocol, that will allow off-chain consensus. Thanks to the Proof-of-Contribution, external resource providers will have the usage of their resources certified directly in the blockchain.

iExec aims to deploy a scalable, high-performance, secure and manageable infrastructure sidechain that will promote a new form of distributed governance, involving key HPC, big data and cloud industry leaders.

iExec is using its ERC20-compliant token to provide standard and secure payments. RLC[5] which stands for "Run on Lots of Computers", can be securely and easily stored, transferred, traded, divided and used to make payments. This widely adopted cryptocurrency (87 million RLC are currently in circulation) is used to access all iExec's services.

From a research project, iExec is now a company, whose headquarters are in Lyon, France, with a subsidiary in Hong Kong.

1.4 Problematic

With the huge increase of human dependency on Information Technology for even daily activities, comes the massive consumption of electricity. In 2016, global data centers used roughly 416 terawatts (more than 90 billion kilowatt-hours for just U.S. data centers) or about 3% of the total electricity in terms of percentage, which is nearly 40% more than the entire United Kingdom. Predictions have shown that this consumption will double every four years[6].

As processing-power demanding technologies like artificial intelligence and blockchain have been appearing, the network of data centers that have sprung up in the past decade will spread. Moreover, internet-connected devices is changing the entire landscape because IoT is projected to exceed 20 billion devices by 2020. Given there are currently 10 billion devices, doubling that will require huge increases to our data center infrastructure, which will massively grow our electricity consumption, and that is just adding fuel to the fire.

There has been some trials to remedy the situation such as using other alternatives to silicon in data storage and counting on virtualisation to reduce the use of physical machines, but all of that still can not keep up against the extreme consumption demand.

1.5 Suggested solution

Although the existing trials are aiming to make datacenters greener, in this project we are taking another perspective. The idea is to push computation (or part of it) to the edge of the network instead of sending it to the cloud (fog/edge computing principles) and use solar energy to power devices that execute this computation. iExec plays a key role in this approach because the execution process will be handled by its software, in other words we are creating an iExec's worker that is completely autonomous and energy positive. The prototype of this system is a Raspberry Pi based device powered by a solar panel but the concept can be applied to a wider range of use cases.

1.6 Conclution

After talking about the project's context and background, we presented the host company and discussed the problematic we are dealing with as well as the solution to resolve this issue, which is the subject of this project.

Chapter 2

State of The Art

2.1 Introduction

- what's out there

2.2 Critical overview of the existens

- Energy Consumption - Idle resources

2.3 PoCo

2.4 XtremWeb

2.5 RLC

2.6 Blockchain - Ethereum

2.7 Cloud computing

2.8 IoT

2.9 Fog computing

2.10 Edge computing

2.11

2.12 Conclusion

Chapter 3

Requirements Analysis

3.1 Introduction

This chapter is about the overall vue of the project analysis and requirements. We start with analysing the global system and explaining its basic architecture then we illustrate that with a general sequence diagrams. In a second section we specify the project requirements and give a brief explanation of each one of them.

3.2 Global System Analysis

3.2.1 iExec ecosystem

iExec creates a global and open market where computing power is traded like a commodity. This ecosystem has different components[7]:

- **The user** who issues computing resources to run decentralized applications. He is identified by his wallet and should use RLC to pay for execution.
- **The worker** or resource provider who lends his machine power and monetize it. The worker builds his reputation according to executions he does, he should prove the correctness of the execution in order to get rewarded in RLC.
- **The scheduler** manages the match-making of the demand and supply in the marketplace, does the necessary controls during the different phases of the process and finalizes the payments as well as sending results back to users.
- **The blockchain** is the laying underground so everything runs on top of it. Payments, PoCo, workers' reputations and much more are services guaranteed by the blockchain.

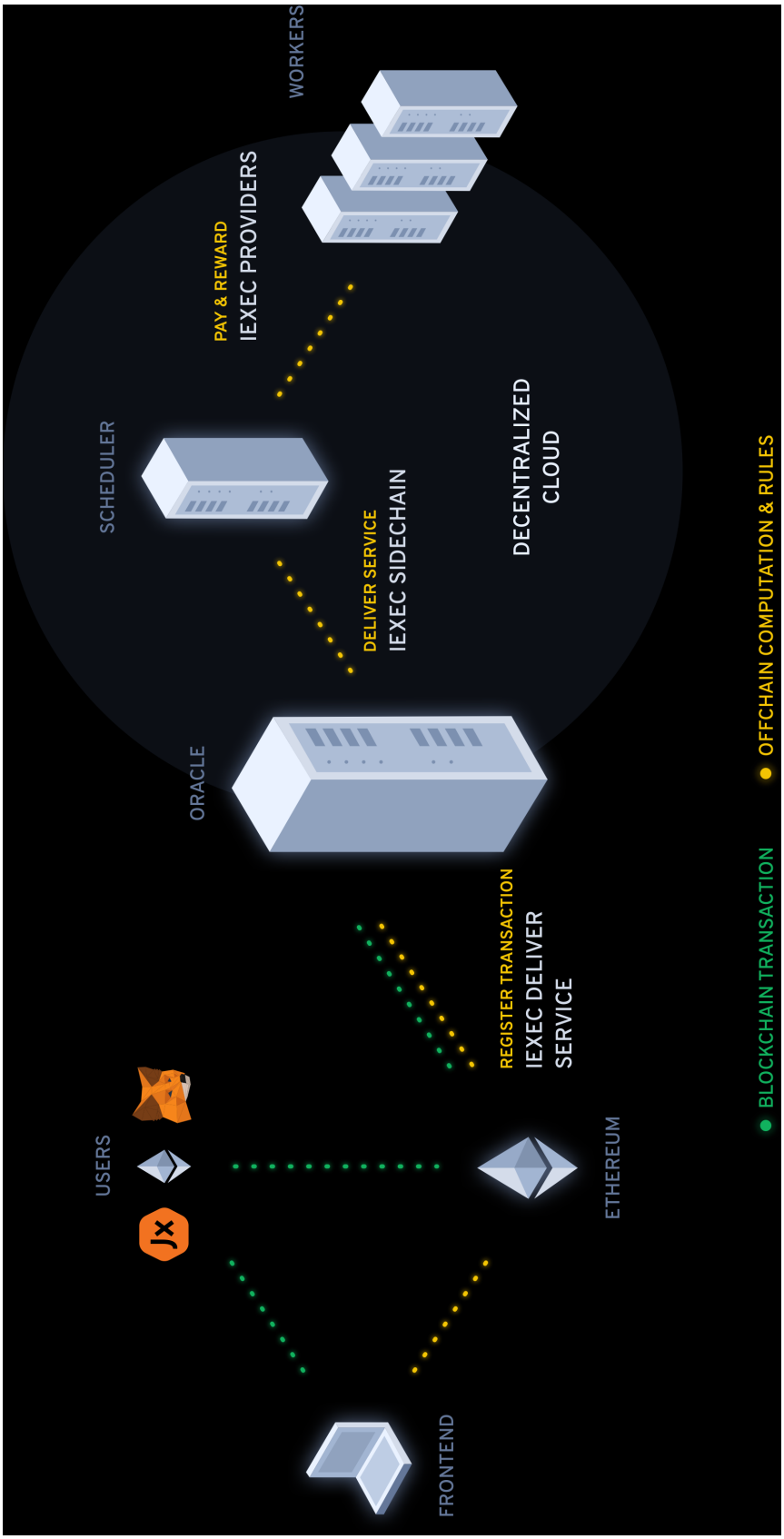

Fig. 3.1 iExec decentralized cloud architecture

The following figure[8] presents the iExec ecosystem and highlights the different parts of it including the marketplace and the worker pools.

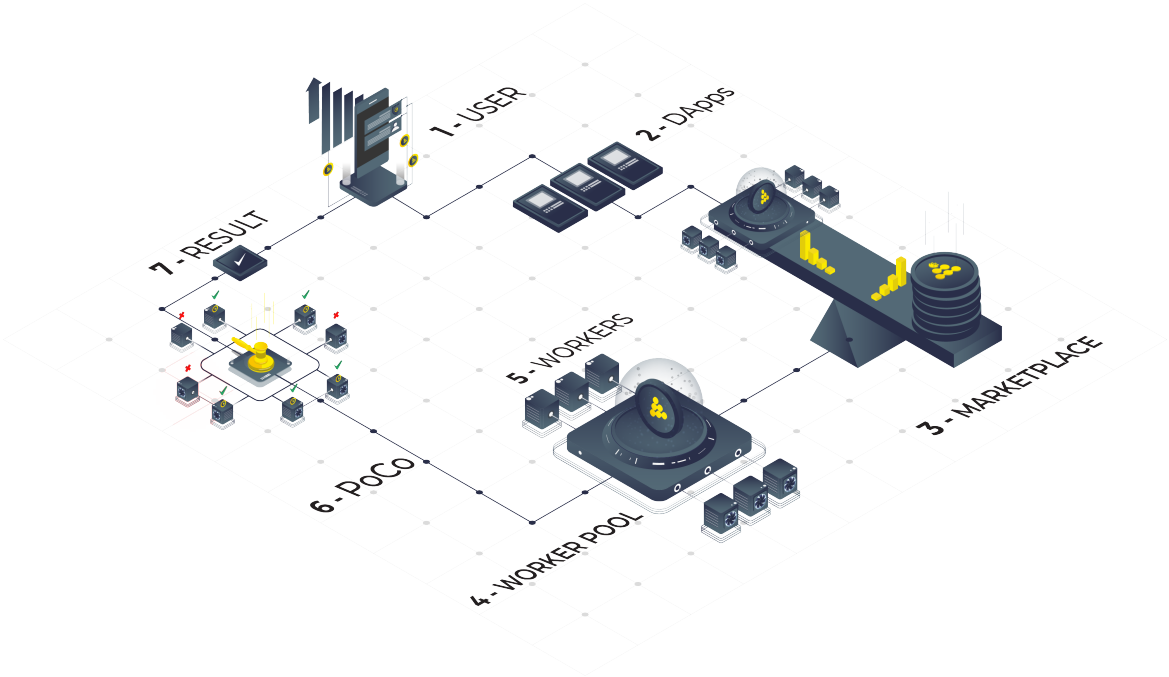

Fig. 3.2 iExec marketplace

3.2.2 General Sequence Diagram

The worker is one of three components of the global architecture, in addition to the scheduler and the user. The user starts the process by issuing and order (sending a job to be executed). The scheduler does the verification of the funds for all parts. It checks the accounts of the user and the worker and insures that they have enough RLC, delegates the job to the worker, gets the result, runs the PoCo to verify the execution, if no problem occurs it finalizes the payment and sends the result back to the user. The worker's mission starts once the job is delegated, it downloads the docker image of the application, downloads also the data needed by the execution, executes the task and uploads the result back to the scheduler. The worker has always to communicate with the scheduler to update his status (alive, working, waiting for work...) and give the scheduler the ability to track the execution process.

The sequence diagram illustrates the details of all the process:

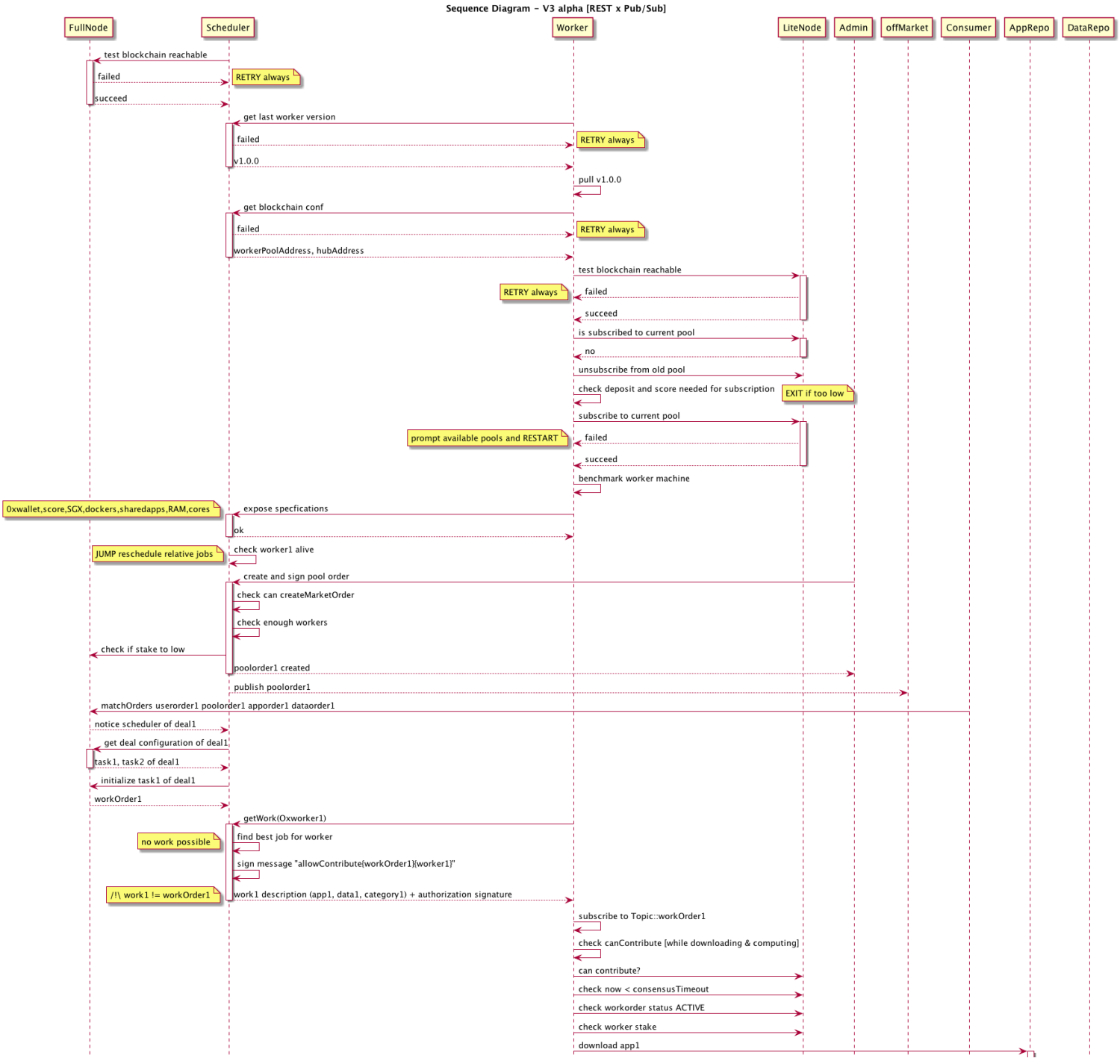

Fig. 3.3 iExec core sequence diagram - part 1

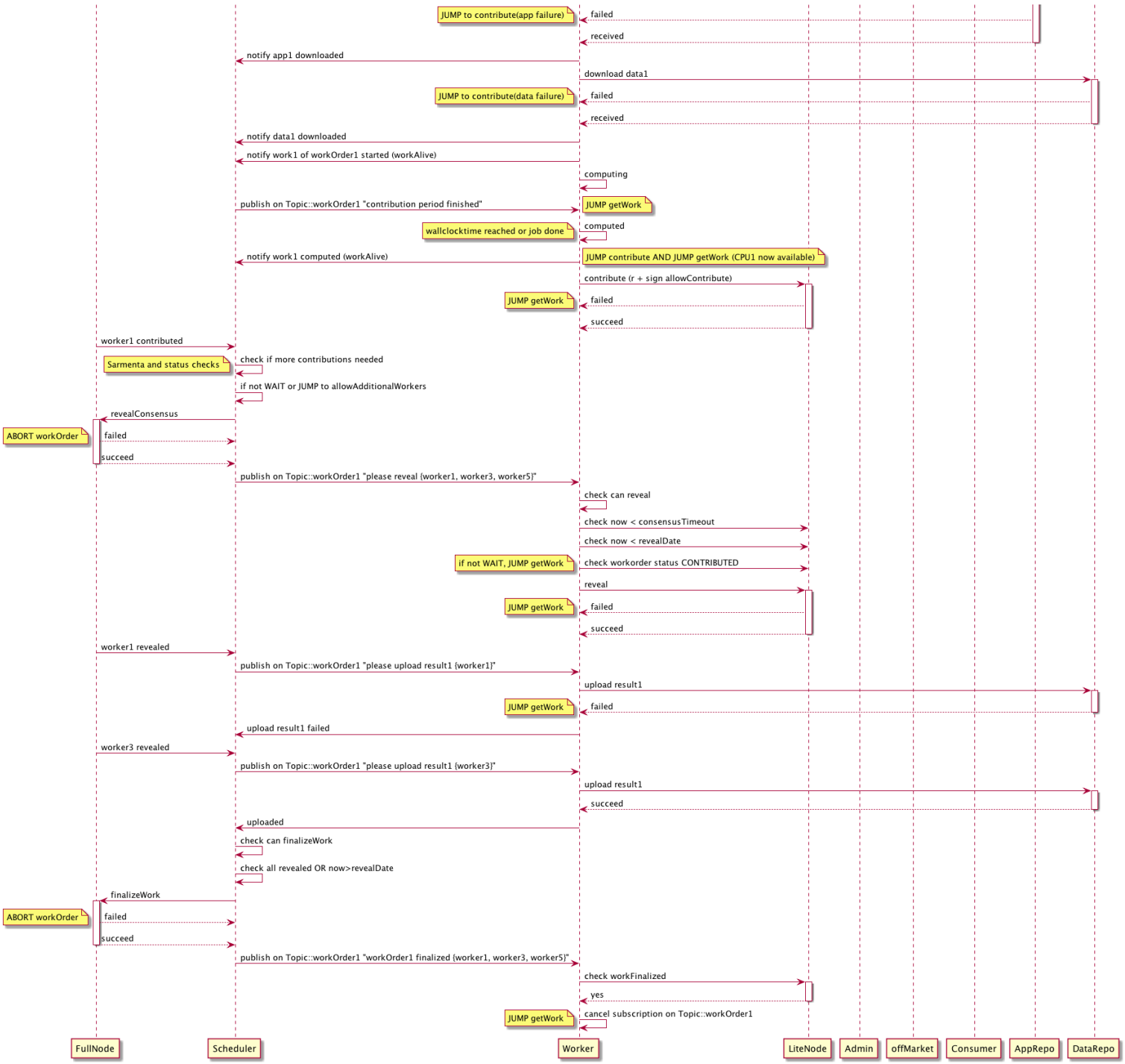

Fig. 3.4 iExec core sequence diagram - part 2

3.3 Requirements

3.3.1 Functional Requirements

The project aims to povide a certain number of functionalities, some of them are absolutely mondatory and some others depend on the use case. The main features are listed below.

- **Execute jobs sent by the middleware:** The worker is a part of iExec's infrastructure, so it should execute any type of job sent by XtremWeb middleware, the source of these tasks is blockchain dapps that need computation resources. The only constraint that those tasks should repect is the specs limitation of the device, Raspberry Pi cards for instance. It's the responsibility of the dapps developer and the task sender to deal with this contraint.

- **Ability to prove execution:** The proof-of-contribution[4] is the protocole that verifies the execution of tasks, so the worker should be able to contribute to the process of the verification. In order to achieve that, the worker should have an identity represented by its blockchain wallet and its account in the iExec ecosystem.

- **Accomplish the mission of an IoT device:** The worker is always available to execute tasks sent by users, but it can also be considered as an IoT device that processes/sends data collected by its sensors. For instance, the worker can be a surveillance camera and analyse the video stream to detect motions or a weather station that collects informations about the temprature, the humidity, the wind speed ...etc.

- **Ability to manage a cluster of workers:** The project is meant to be applicable on a large number of workers, so it is hard to manage them seperatly, that is why it is necessary to have a way to manage them easily as a cluster.

3.3.2 Non-Functional Requirements

The project emphasizes some extremely important non-functional requirements. Those key specifications should be respected in order to maintain the spirit of the project.

- **Positive energy:** One of the main features of this project is to eliminate the energy cost and design a system that is totally green and nature friendly. The worker is powered by solar energy and would produce more energy than it consumes. The electricity is saved in a battery to avoid climate's change effects.

- **Support ARM architecture:** Who says IoT says ARM[9] because this architecture is the omnipresent architecture in the IoT ecosystem, so the project has absolutely to support it. The challenge would be to build binaries for ARM using non-ARM hardware.

3.4 Conclusion

After understanding the global architecture of the iExec environment and defining the roles of the different components, we identified the functional and non-functional requirements of the project. Those requirements would lead us to the next part where we design our system and detail its technical specifications.

Chapter 4

Design & technical specifications

4.1 Introduction

Once the Requirements are specified

4.2 Desing and architecture

4.2.1 Global System Architecture

4.2.2 Scheduler

4.2.3 Worker

4.3 Software Development Kit (SDK)

4.4 Detailed Design

4.5 Technical Specifications

4.5.1 concepts to use

4.5.2 automate configuration and deployment

4.5.3 develop dapps (use cases)

4.5.4 deployment of all the environment

4.5.5 cross-compiling

4.6 Conclusion

Chapter 5

Implementation

5.1 Introduction

- intro

5.2 Work environment

5.2.1 Hardware environment

5.2.2 Electronic Card: Raspberry Pi

5.2.3 Solar Panel

5.2.4 Battery

5.2.5 Witty Pi

5.2.6 Software environment

5.2.7 Development Technologies

5.2.8 Documentation

5.2.9 cluster + automation

5.3 Illustration

5.4 Conclusion

General Conclusion And Perspectives

References

[1] Tom Bawden. Global warming: Data centres to consume three times as much energy in next decade, experts warn. *Independent*, January 2016. https://www.independent.co.uk/environment/global-warming-data-centres-to-consume-three-times-as-much-energy-in-next-decade-experts-warn-a6830086.html. Last access date: 24/08/2018 11:00.

[2] iExec https://www.iex.ec.

[3] XtremWeb https://www.xtremweb-hep.lal.in2p3.fr/.

[4] PoCo https://www.iex.ec/news_categories/poco-series/.

[5] RLC token https://www.iex.ec/rlc-token/.

[6] Radoslav Danilak. Why energy is a big and rapidly growing problem for data centers. *Forbes*, December 2017. https://www.forbes.com/sites/forbestechcouncil/2017/12/15/why-energy-is-a-big-and-rapidly-growing-problem-for-data-centers/#682a483c5a30. Last access date: 24/08/2018 11:00.

[7] iExec decentralized cloud https://www.iex.ec/overview/.

[8] iExec marketplace https://www.iex.ec/marketplace/.

[9] ARM architecture https://www.arm.com.