Positive Energy Worker

Autonomous Positive Energy Machine Running on The Blockchain

Zied Guesmi

Supervisors: Prof. A.B. Supervisor
Prof. C.D. Supervisor

Advisors: Dr. A. Advisor
Dr. B. Advisor

Ministry of Higher Education And Scientific Research.
University of Tunis ELMANAR, Tunisia
Faculty of Mathematical, Physical and Natural Sciences of Tunis
Department of Computer Science Engineering

This dissertation is submitted for the degree of
Computer Engineer

Department of Computer Science
Engineering

September 2018

Dedication . . .

Acknowledgements

And I would like to acknowledge ...

Abstract

This is where you write your abstract ...

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

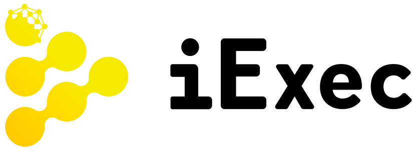

Fig. 1.1 iExec logo

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

Before elaborating our solution, we discuss in this section some existing approaches and their limitations. We define, after, multiple technologies and concepts we use in this project.

2.2 Critical overview of the existing

There has not been other trials that use the same approach as this project, but there are some other attempts to remedy the problem of energy consumption in the IT field. Those solution suggest that datacenters should be powered using renewable energy sources.

2.2.1 Solar-Powered Data Centers

Emerson Network Power, i/o Data Centers and AISO are among the companies that have implemented on-site solar solutions. Emerson Network Power has installed a 7,800 square foot solar array on the roof of its new St. Louis data center[7].

Fig. 2.1 Emerson Network Power solar array

Also, QTS Princeton data center campus installed more than 57,000 solar panels in New Jersey countryside. The installation generates up to 14.1 megawatts of power. That's more than enough to supply the daytime energy needs of the data center[8].

Fig. 2.2 QTS Princeton datacenter photovoltaic solar panels

2.2.2 Wind-Powered Data Centers

Only a handful of companies have implemented wind turbines in working data centers. A small ISP and hosting company in Woodstock, Illinois, Other World Computing (OWC) may be the first data center operator in the U.S. to power its facility entirely with wind power from an on-site turbine. In 2009, OWC began using a 131-foot-tall wind turbine to provide all the electric power for its building, which includes the company's headquarters and a data center supporting its web hosting and ISP services[9].

Fig. 2.3 The 131-foot tall wind turbine at Other World Computing in Woodstock, Illinois

2.2.3 Geothermal Data Centers

A number of big companies like Google and Facebook are using geothermal cooling in their data centers. They have built them in icy areas like Sweden and Finland taking advantage of the cold air in those regions.

2.2.4 Limitations of previous solutions

Even though most of these approaches exist in real world, they still have problems reaching the scale required to successfully support the power requirements of big data centers. Solar power hasn't been widely used in data centers because it takes a very large installation of photovoltaic (PV) solar panels to produce even a fraction of the energy required by most data centers. In addition to that, the cost of deploying such solutions is still exorbitant.

Furthermore, working on the datacenter itself does not take advantage of the vast network of connected devices all over the world and keeps these important resources idle and not effectively used.

2.3 State of the art

In this section, we explain some key concepts and technologies we used during this project. Some of them are distributed systems related and some others describe some blockchain notion.

- **Ethereum blockchain:**

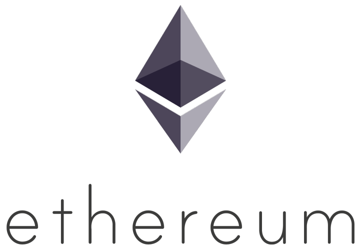

Fig. 2.4 Ethereum blockchain

Ethereum[10] was initially described by Vitalik Buterin[11] in late 2013 as a result of his research and work in the Bitcoin community. Vitalik published the Ethereum white paper, where he describes in detail the technical design and rationale for the Ethereum protocol and smart contracts architecture. In January 2014, Ethereum was formally announced by Vitalik at the The North American Bitcoin Conference in Miami, Florida, USA.

It is an open Blockchain platform that lets anyone build and use decentralized applications. Like Bitcoin, no one controls or owns Ethereum – it is an open-source project built by the community around the world. But unlike the Bitcoin protocol, Ethereum was designed to be adaptable and flexible. It is easy to create new applications on the Ethereum platform.

Ethereum is a programmable Blockchain. Rather than giving users a set of predefined operations (e.g. Bitcoin transactions), Ethereum allows users to create their own operations of any complexity they wish. In this way, it serves as a platform for many different types of decentralized applications, including but not limited to cryptocurrencies.

So basically, ethereum allows exchanging ether which makes it behave like Bitcoin does. However, ethereum allows anybody to write any piece of code (Smart Contract) and upload it to the Blockchain so anyone can interct with it which brings us to cite that there is two types of accounts that lives within the ethereum Blockchain: externally owned accounts which are controlled by private keys and contrat accounts which are controlled by a piece of code.

It is the Ethereum Virtual Machine (EVM) that executes the code of smart contracts and it is generally expensive to run applications on top of ethereum because of gaz price, so iExec's plateform takes this execution offchain in order to allow on-demand, secure and low-cost access to competitive computing infrastructures.

- **XtremWeb[3]:** a data driven volunteer cloud middleware used by iExec to manage the offchain computations. The middleware implements many features needed such as authentication, tasks scheduling, fault-tolerance, security (...).
  It was mainly developed by Oleg Lodygensky[12] at the LAL laboratory in IN2P3 based on XtremWeb 1.8.0 by INRIA.

- **RLC[5]:**

Fig. 2.5 RLC token

It is the token used to access the resources provided by the market network. It is the unique way of payment for application providers, server providers and data providers.

- **iExec Software Development Kit[13] (SDK):** this tool gives the ability to deploy any legacy applications in the iExec infrastructure, and execute them through calls to Ethereum smart contracts.

- **PoCo:** the Proof-of-Contribution protocol[4] ensures that the results computed by the workers are valid and can be trusted by the user who required them. It controls the way multiple workers achieve consensus on a computation result by leveraging several mechanisms[14]. This protocol was developed by Hadrien Croubois[15] a Ph.D. student at ENS Lyon, France.

- **IoT:** The internet of things, or IoT, is a system of interrelated computing devices, mechanical and digital machines, objects, animals or people that are provided with unique identifiers (UIDs) and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction[16].

- **Fog computing:** Fog computing, also known as fog networking or fogging, is a decentralized computing infrastructure in which data, compute, storage and applications are distributed in the most logical, efficient place between the data source and the cloud. Fog computing essentially extends cloud computing and services to the edge of the network, bringing the advantages and power of the cloud closer to where data is created and acted upon[17].

- **Edge computing:** Edge computing in IT is defined as the deployment of data-handling activities or other network operations away from centralized and always-connected network segments, and toward individual sources of data capture, such as endpoints like laptops, tablets, smartphones or IoT devices[18].

- **photovoltaic system:** A photovoltaic system, also PV system or solar power system, is a power system designed to supply usable solar power by means of photovoltaics. It consists of an arrangement of several components, including solar panels to absorb and convert sunlight into electricity, a solar inverter to change the electric current from DC to AC, as well as mounting, cabling, and other electrical accessories to set up a working system.

- **Power system:** An electric power system is a network of electrical components deployed to supply, transfer, and use electric power. An example of an electric power system is the grid that provides power to an extended area. An electrical grid power system can be broadly divided into the generators that supply the power, the transmission system that carries the power from the generating centres to the load centres, and the distribution system that feeds the power to nearby homes and industries https://www.allaboutcircuits.com/ [Online textbook], Tony R. Kuphaldt et al., last accessed on 17 May 2009.

- **Solar Power:** Solar power is the conversion of energy from sunlight into electricity, either directly using photovoltaics (PV), indirectly using concentrated solar power, or a combination. Concentrated solar power systems use lenses or mirrors and tracking systems to focus a large area of sunlight into a small beam. Photovoltaic cells convert light into an electric current using the photovoltaic effect https://www.energy.gov/science-innovation/energy-sources/renewable-energy/solar Department of Energy. Archived from the original on 14 April 2011. Retrieved 19 April 2011.

- **Photovoltaics:** Photovoltaics (PV) is the conversion of light into electricity using semiconducting materials that exhibit the photovoltaic effect, a phenomenon studied in physics, photochemistry, and electrochemistry. A typical photovoltaic system employs solar panels, each comprising a number of solar cells, which generate electrical power. PV installations may be ground-mounted, rooftop mounted or wall mounted. The mount may be fixed, or use a solar tracker to follow the sun across the sky. Lo Piano, Samuele; Mayumi, Kozo (2017). "Toward an integrated assessment of the performance of photovoltaic power stations for electricity generation". Applied Energy. 186 (2): 167–74.

- **Solar panel:** Photovoltaic solar panels absorb sunlight as a source of energy to generate electricity. A photovoltaic (PV) module is a packaged, connected assembly of typically 6x10 photovoltaic solar cells. Photovoltaic modules constitute the photovoltaic array of a photovoltaic system that generates and supplies solar electricity in commercial and residential applications. Ulanoff, Lance (2 October 2015). "Elon Musk and SolarCity unveil 'world's most efficient' solar panel". Mashable. Retrieved 9 September 2018

- **photovoltaic system:**

2.4 Conclusion

iExec ecosystem provides an extremely important plateform and set of tools that serves perfectly the purpose of this project. Implementing the Energy Positive Worker would make use of the architecture and the business model to find the incentive to push this project toward a bright future.

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. Its uses the blockchain as the laying underground so everything is certified on top of it. This guarantees multiple services such as the payments and the PoCo (...). This plateform has different agents[19]:

- **The user** who issues computing resources to run decentralized applications. He is identified by his wallet and uses RLC to pay for the rented resources.
- **The worker** or resource provider who lends his machine power and monetize it. The worker builds his reputation according to executions he does.
- **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 application developer** designs applications running on the Ethereum[10] blockchain.

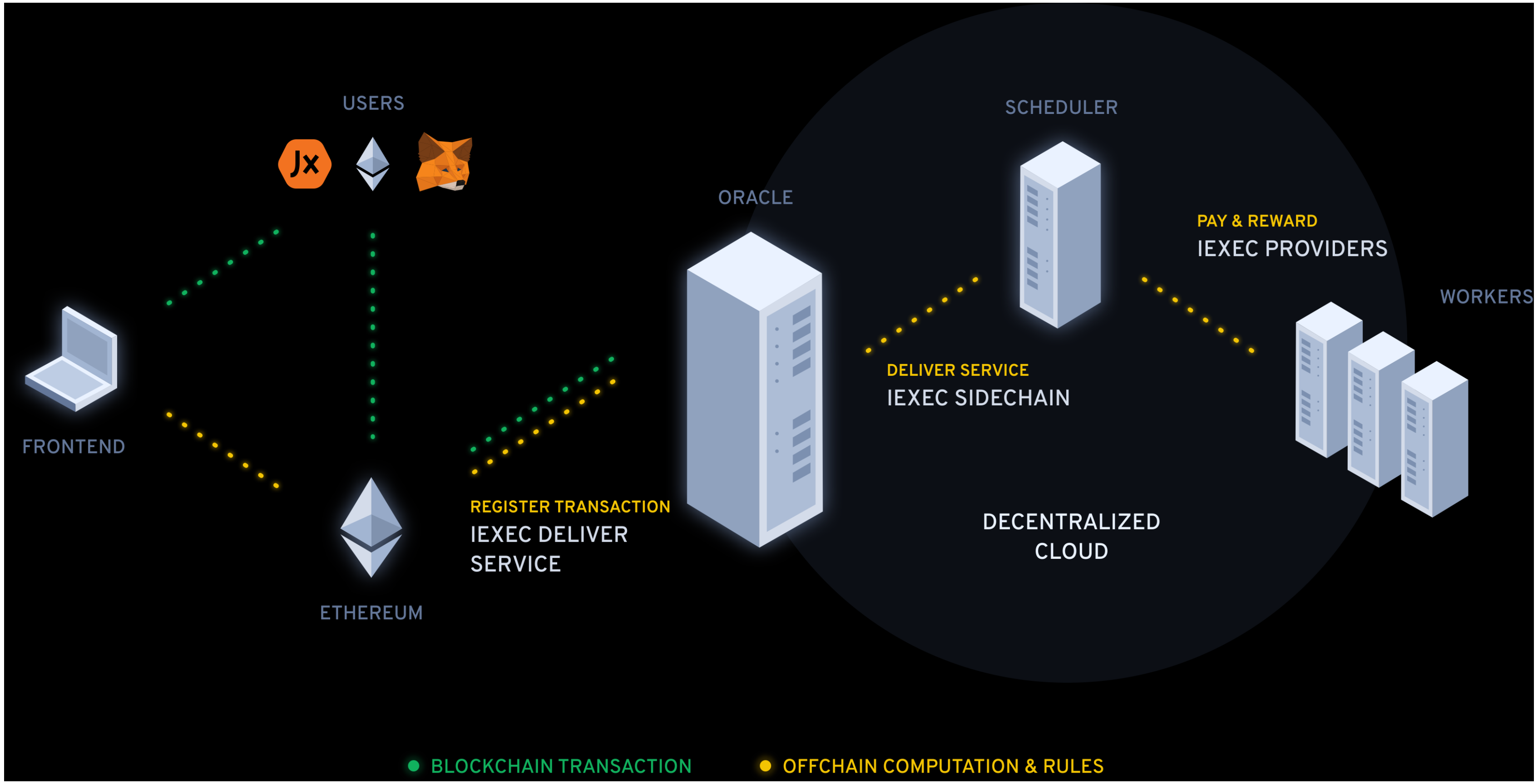

Fig. 3.1 iExec decentralized cloud architecture

The following figure[20] presents the iExec ecosystem and highlights the different parts of it including the marketplace and the worker pools.

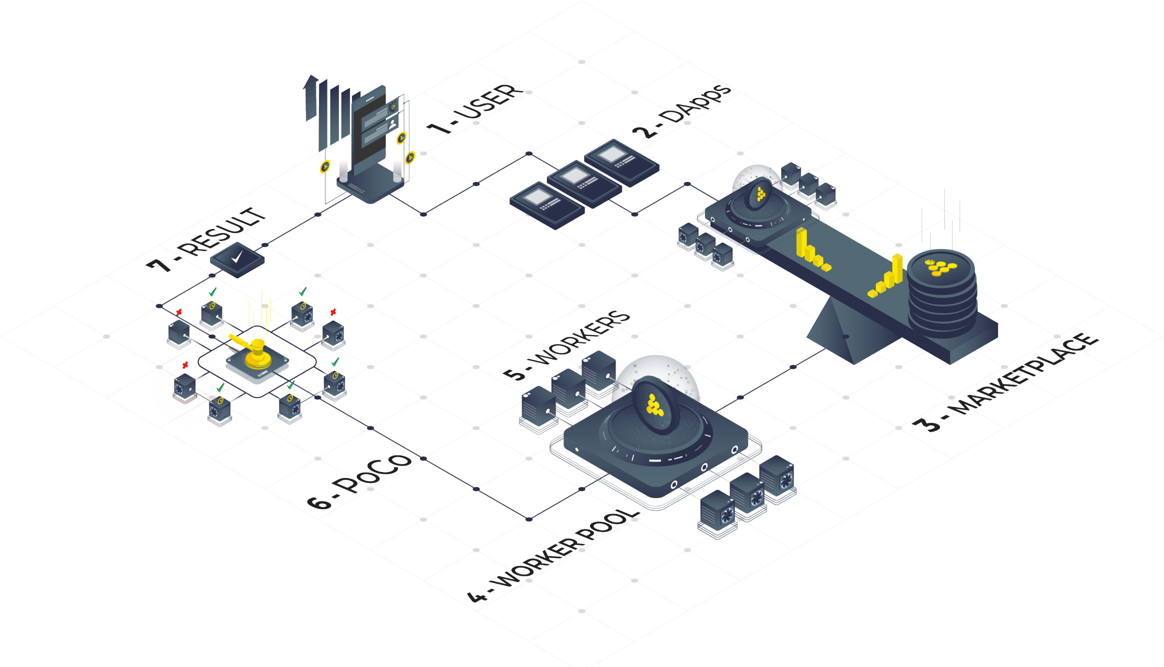

Fig. 3.2 iExec marketplace

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:

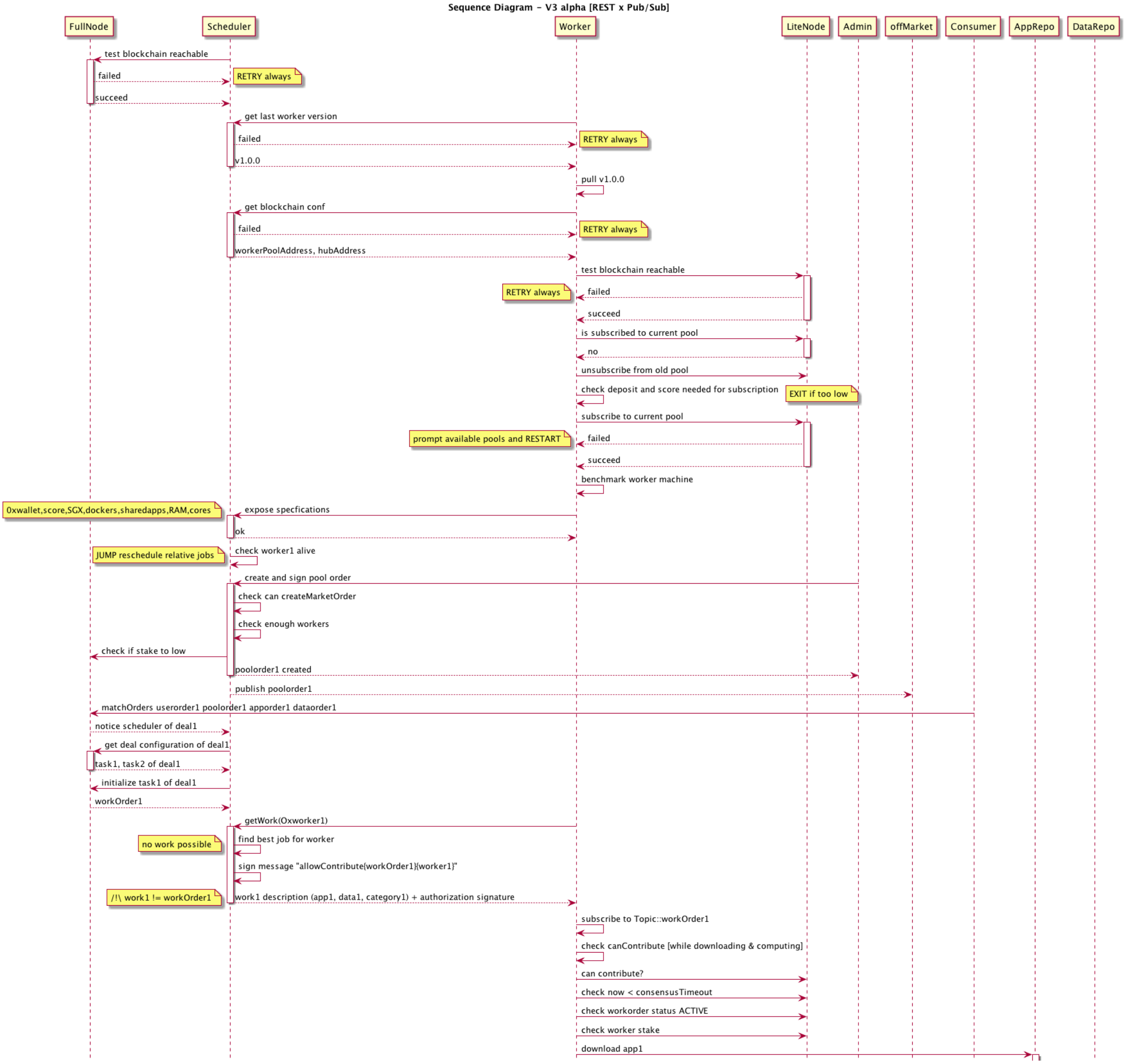
Sequence Diagram - V3 alpha [REST x Pub/Sub]
FullNode
Scheduler
Worker
LiteNode
Admin
offMarket
Consumer
AppRepo
DataRepo
test blockchain reachable
failed
RETRY always
succeed
get last worker version
failed
RETRY always
v1.0.0
pull v1.0.0
get blockchain conf
failed
RETRY always
workerPoolAddress, hubAddress
test blockchain reachable
RETRY always
failed
succeed
is subscribed to current pool
no
unsubscribe from old pool
check deposit and score needed for subscription
EXIT if too low
subscribe to current pool
prompt available pools and RESTART
failed
succeed
benchmark worker machine
0xwallet,score,SGX,dockers,sharedapps,RAM,cores
expose specfications
ok
JUMP reschedule relative jobs
check worker1 alive
create and sign pool order
check can createMarketOrder
check enough workers
check if stake to low
poolorder1 created
publish poolorder1
matchOrders userorder1 poolorder1 apporder1 dataorder1
notice scheduler of deal1
get deal configuration of deal1
task1, task2 of deal1
initialize task1 of deal1
workOrder1
getWork(0xworker1)
no work possible
find best job for worker
sign message "allowContribute(workOrder1)(worker1)"
/!\ work1 != workOrder1
work1 description (app1, data1, category1) + authorization signature
subscribe to Topic::workOrder1
check canContribute [while downloading & computing]
can contribute?
check now < consensusTimeout
check workorder status ACTIVE
check worker stake
download app1

Fig. 3.3 iExec core sequence diagram - part 1

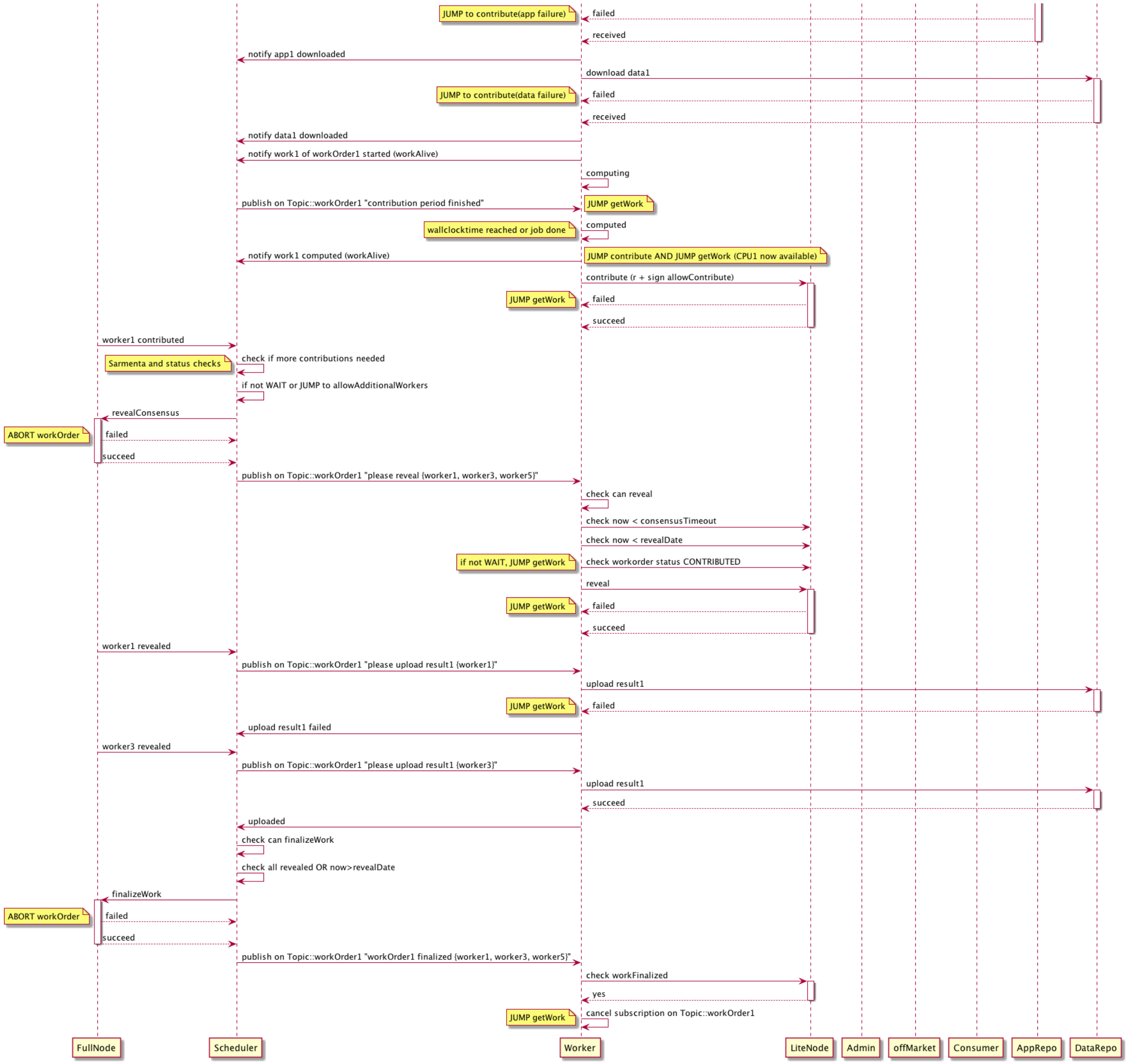
JUMP to contribute(app failure)
failed
received
notify app1 downloaded
download data1
JUMP to contribute(data failure)
failed
received
notify data1 downloaded
notify work1 of workOrder1 started (workAlive)
computing
publish on Topic::workOrder1 "contribution period finished"
JUMP getWork
wallclocktime reached or job done
computed
notify work1 computed (workAlive)
JUMP contribute AND JUMP getWork (CPU1 now available)
contribute (r + sign allowContribute)
JUMP getWork
failed
succeed
worker1 contributed
Sarmenta and status checks
check if more contributions needed
if not WAIT or JUMP to allowAdditionalWorkers
revealConsensus
ABORT workOrder
failed
succeed
publish on Topic::workOrder1 "please reveal {worker1, worker3, worker5}"
check can reveal
check now < consensusTimeout
check now < revealDate
if not WAIT, JUMP getWork
check workorder status CONTRIBUTED
reveal
JUMP getWork
failed
succeed
worker1 revealed
publish on Topic::workOrder1 "please upload result1 {worker1}"
upload result1
JUMP getWork
failed
upload result1 failed
worker3 revealed
publish on Topic::workOrder1 "please upload result1 {worker3}"
upload result1
succeed
uploaded
check can finalizeWork
check all revealed OR now>revealDate
finalizeWork
ABORT workOrder
failed
succeed
publish on Topic::workOrder1 "workOrder1 finalized {worker1, worker3, worker5}"
check workFinalized
yes
JUMP getWork
cancel subscription on Topic::workOrder1
FullNode
Scheduler
Worker
LiteNode
Admin
offMarket
Consumer
AppRepo
DataRepo

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. The objectif is to charge the battery with enough electricity for two days.

- **Support ARM architecture:** Who says IoT says ARM[21] 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 & Hardware

4.1 Introduction

Once the system requirements are identified, the next step would be to prepare the conceptual representation of the project. This part focuses just on the worker, specifying its requirements and going all the way from the software and the communication with the scheduler to the hardware specifications.

4.2 Desing & Technical Specifications

The following diagram presents the interaction between the different components of the worker. First of all, the connection between the components is guaranteed by the TCP protocol. The ports 4321, 4322, 4323 and 4324 should be open in both machines, the worker and the scheduler. At the start-up, the worker downloads the CA certificate. This part is mandatory for the program to continue working.

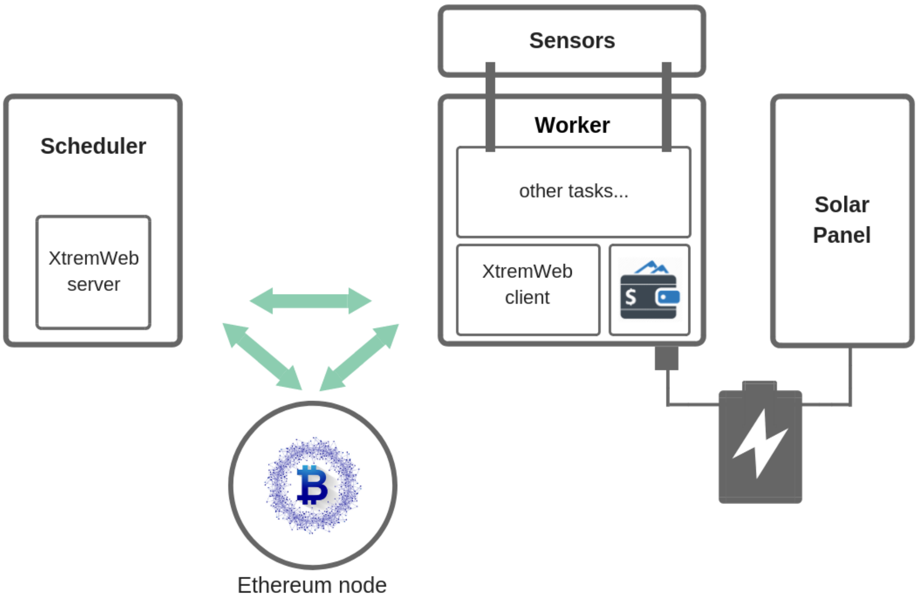

Fig. 4.1 Global overview of the worker design

We explain in the following subsections the role and requirements of each component.

4.2.1 Worker

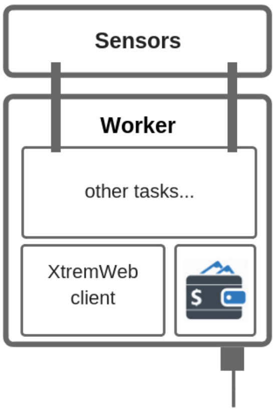

Fig. 4.2 The worker internal sortware components

Fig. 4.3 The worker internal sortware components

The worker runs a docker image of XtremWeb worker. This image has to be optimized for devices with limited resources like the Raspberry Pi board. XtremWeb does not support ARM architecture natively so this support should be introduced to the new version of the middleware. Luckily docker is available for this architecture which simplifies the task.

4.2.2 Blockchain

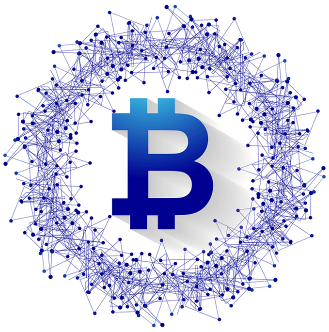

Fig. 4.4 The solar panel

The blockchain part is represented by the Geth node where we deploy the iExec smart contracts. The worker should take care of the necessary blokchain configuration for the proper functionning of the system, so it should provide a wallet with sufficient funds (in RLC). The scheduler locks an amount of RLC when an execution is starting, so the worker should have the minimum amount required by the operation. If these requirements are not satisfied the program will crash.

4.2.3 Use case diagram

TODO

4.3 Hardware choice

4.3.1 Solar Panel

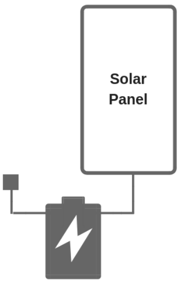

Fig. 4.5 The solar panel

A Photovoltaic solar panels is used to power the worker. It absorbs sunlight as a source of energy to generate electricity and charge the battery. In this case we use a panel with high conversion efficiency: 60 watt SUNPOWER mono-crystalline with conversion efficiency up to 25% which is much higher than common solar panel charger (15%).

Reference of the chosen panel: SUAOKI 60W Portable Sunpower Mono-crystalline Solar Panel with DC 18V and USB 5V Output Charger.
Output: up to 60W - USB: 18v/3.4A, DC: 5v/2.5A.

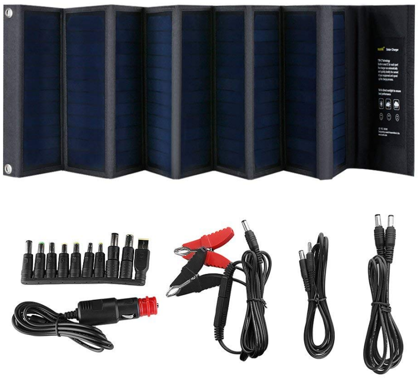

Fig. 4.6 SUNPOWER solar panel

This panel should be able to charge the battery in merely 8 hours during a sunny day. The Raspberry Pi board consumes 1A for a maximum utilization of the CPU and some plugged sensors. For a battery capacity of 50000mA, the worker can stay up for 24 hours for a full battery without any source of solar light.

4.3.2 Battery

Fig. 4.7 50000 mA Battery

We suppose that the battery should be able to store electricity for, at least, two days. With a consumption of maximum 24A/day, the battery should store 50A. It should also contain a 5V output to power the worker with the right intensity.

Reference of the chosen battery: MAXOAK 50000mAh 5/12/20v Portable Charger External Battery Power Bank.

4.3.3 Sensors

The worker uses different sensors to interact with the outside world. We can imagine multiple systems with a lot of use cases. We will take as example two use cases:

Surveillance camera: the worker has an 8 megapixel native resolution sensor-capable of 3280 x 2464 pixel static images camera to achieve the mission of a surveillance system that streams video and detects motions.

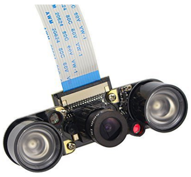

Fig. 4.8 Raspberry Pi camera

Weather station: in this case the worker uses a humidity, pressure, temprature and orientation senors to build a weather station. It collects weather data and applies some processing on it before sending it to a weather forcast service. The data processing can be done with an iExec dapp locally on the worker and this model has many advantges such as pushing the processing to the edge of the network to reduce latency caused by data transfer and enhance data privacy because it is not leaving the device.

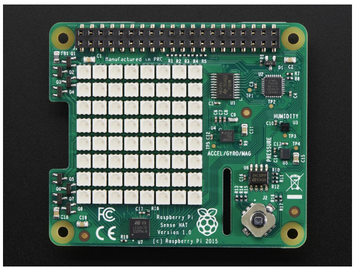

Fig. 4.9 Raspberry Pi Sense HAT

4.4 Conclusion

In this chapter, we worked on the first place on designing the worker taking in consideration the resources limitation of the worker board. In the second section, we worked on choosing the hardware that suits the requirements, therefore we pass to the implementation part to shape our design.

Chapter 5

Implementation

5.1 Introduction

After fixing the system design and choosing the apropriate hardware, it is time to implement the worker. The Implementation consists of two parts, assembling the hardware components and preparing the software. To test the worker we develop some dapps for ARM architecture and integrate them in the iExec plateform. In the last section we illustrate the work that has been done with some pictures of the worker prototype.

5.2 Work environment

- Amazon AWS instance as a scheduler and a blockchain node
- 3 Raspberry Pi boards as a cluster of workers
- A laptop as a client to test the infrastructure

5.3 Software environment

5.3.1 Blockchain

Generate wallets using the SDK

5.3.2 XtremWeb

add arm support to the scheduler - cross compiling resin + qemu

5.3.3 Docker

5.3.4 Worker cluster

cluster –> swarm - cluster - ansible to verify

5.4 Dapps to test the infrastructure

5.5 Documentation

confluence

5.6 Illustration

5.7 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] Solar-Powered Data Centers https://www.datacenterknowledge.com/solar-powered-data-centers/.

[8] Solar-Powered Data Centers https://www.datacenterknowledge.com/archives/2014/10/14/super-sizing-solar-power-for-data-centers.

[9] Wind-Powered Data Centers https://www.datacenterknowledge.com/wind-powered-data-centers/.

[10] Ethereum blockchain https://www.ethereum.org/.

[11] Vitalik Buterin, Ethereum creator https://about.me/vitalik_buterin.

[12] Oleg Lodygensky, XtremWeb main developer https://users.lal.in2p3.fr/lodygens/.

[13] iExec SDK https://github.com/iExecBlockchainComputing/iexec-sdk.

[14] PoCo definition https://www.medium.com/iex-ec/poco-series-2-on-the-use-of-staking-to-prevent-attacks-2a5c700558bd.

[15] Hadrien Croubois, PoCo designer https://www.hadriencroubois.com/.

[16] What is IoT https://www.internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT.

[17] What is Fog computing https://www.techopedia.com/definition/32472/edge-computing.

[18] What is Edge computing https://www.techopedia.com/definition/32472/edge-computing.

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

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

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