DPS
Decentralized Publisher-Subscriber Protocol

Luca Carabetta
luca@efesto.io

June 10TH, 2017

Abstract

Publisher-Subscriber (Pub-Sub) protocols are widely spread in nowadays technology. Because of their flexibility they could be easily implemented in a vast set of heterogeneous environments. Today this kind of implementations rely on centralized infrastructures, based on broker services and databases hosted on servers. With blockchain technology it's possible to redesign this concept following the paradigms of decentralization, translating into its logic Smart Contracs. The role of this whitepaper is to share the idea of a Decentralized Pub-Sub protocol involving the open-source community into the advanced development of it.

0 Index

0 Index 2
1 Introduction 3
2 Technical deepening 4
2.1 Pub-Sub protocols 4
2.1 DPS: the basics 5
2.3 Exchanges and Queues 6
2.3.1 Exchanges and Queues private/public access 6
2.4 Binding Keys 7
2.5 Routing process 7
2.5.1 Direct delivery 8
2.5.2 Fanout delivery 8
2.5.3 Topic delivery 9
3 Possible further developments 10
3.1 Release conditions for the Queues 10
3.2 Regexp engine 10
3.3 Payments 10
3.4 Hierarchical Routing Keys 11
4 DPS-based applications 12
4.1 Internet of Things 12
4.2 Democracy and organizations 13
4.3 Energy, water and natural gas distribution 14
4.4 Medical, healthcare 15
4.5 Supply chain, logistics, shipments 15
4.6 Messaging, advertisement, people engagement 15
5 Conclusions 16

1 Introduction

The author's main objective regarding this whitepaper is to share an idea to the worldwide community of innovators and then to stimulate the birth of new decentralized projects in the most unexpected areas.

We are now in a very dynamic moment, many consider this period as the one of the birth of the web and it's innovators's responsability to conduct the world to the future, building it brick after brick.

The two main reasons why DPS came out rely on the wide use of pub-sub protocols by the author's works in the IoT and on his passion and curiosity for technology in general and in blockchain in particular.

Pub-Sub is an already well known paradigm and its decentralized translation would improve it in terms of security, anonymity and trust because of public and direct transaction, introducing also unexpected advances. Blockchain technologies are today constantly improving so that in a very near future decentralized applications will disrupt almost any area, overcoming traditional systems in terms of scalability and professional qualities also.

This whitepaper is structured in three parts:

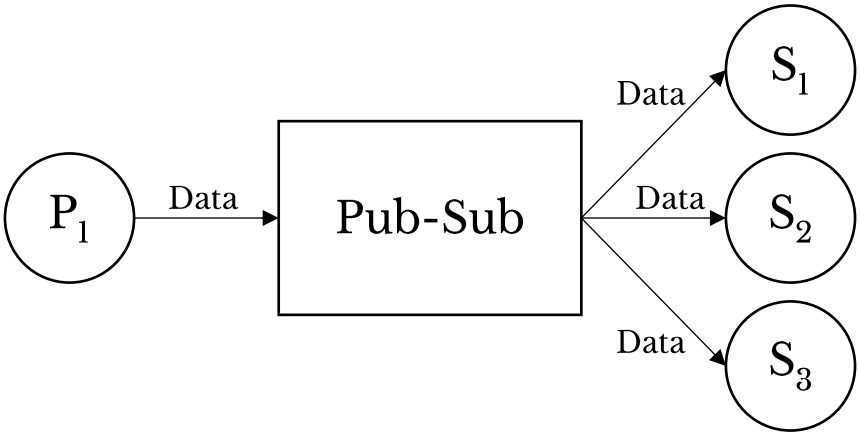
- a technical deepening on DPS;
- suggestions for further developments of DPS;
- suggestions for new applications based on DPS.

This whitepaper and the DPS Smart Contract have to be consider as elements of discussion, anyone can improve them introducing new ideas and features.

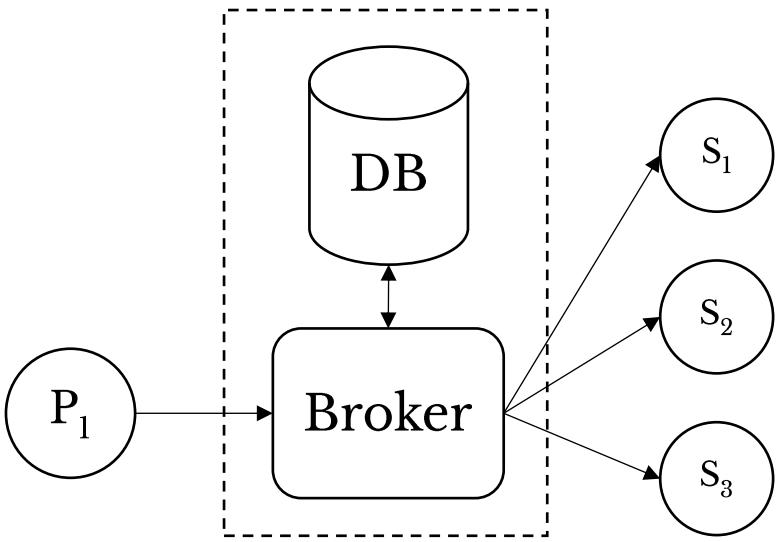
2 Technical deepening

2.1 Pub-Sub protocols

The basic concept in pub-sub protocols is that a Publisher entity (*P*) can deliver data to a set of Subscriber (*S*) entities. Pub-Sub technologies also give utilities to easily route data within the network.



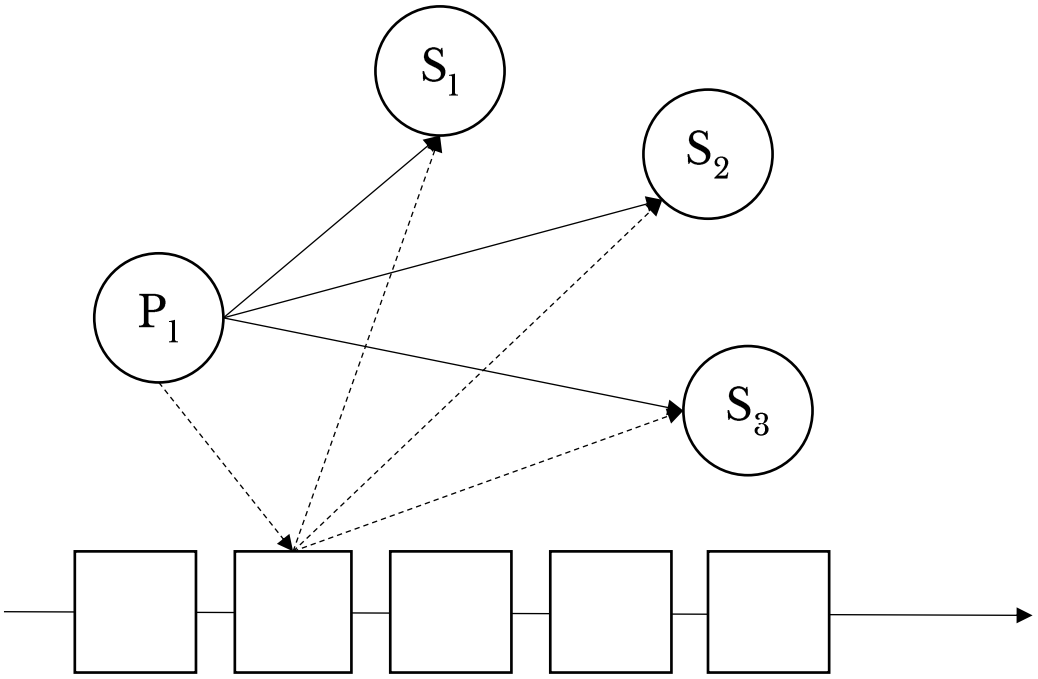
Publisher-Subscriber or Producer-Consumer relationships are fundamental for the world today and this leads to the nowdays scenario in which many applications are based on Pub-Sub protocols built on a centralized infrastructure.



The common setup of a centralized Pub-Sub application involves the use of server (and cloud) services, for instance the simplest implementation would be based on a broker server and a database to store data.

2.1 DPS: the basics

DPS translates all of the well known paradigms of pub-sub protocols in the decentralized world through the use of Smart Contracts on blockchain.



Publisher and subscribers interact with each other sending data through the public ledger in which dedicated Smart Contracts are stored.

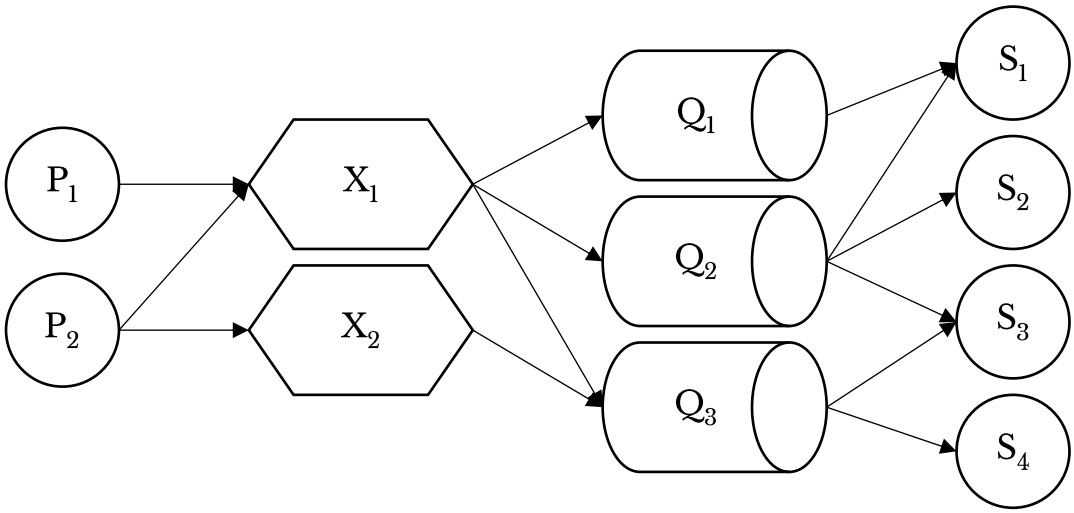
DPS has advantages with respect to other centralized solutions:

- it's safer because of the distributed consensus algorithms which govern blockchain technologies;
- DPS does not involve any broker server or database service because the first role is played by Smart Contracts, data are instead stored within blockchain blocks;
- costs will matter because in DPS there is no need to maintain an infrastructure;
- it's a concept appliable to public or private networks and many implementation of pub-sub protocols have already to be discovered;
- blockchain guarantees anonimity.

DPS will also play an important role in blockchain technologies because it introduces a new and more structured way to implement P2P communication: Smart Contracts contain rules to route message through different nodes.

2.3 Exchanges and Queues

The routing logic and the message delivery are considered as two different processes, respectively governed by two entities: the Exchange (X) and the Queue (Q). An X can receive data from many P and send it to many Q, a Q can receive data from many X and deliver messages to many S.
In centralized technologies Exchanges and Queues live on a broker server, in DPS they are shared in the blockchain as independent Smart Contracts.



Exchanges and Queues are bounded with each other thanks to “Binding Keys”. When publishers send data they specify a “Routing Key”.
The routing logic behind DPS relies on the matching of Routing Keys and Binding Keys; in this way a publisher could easily reach many subscribers with one transaction.

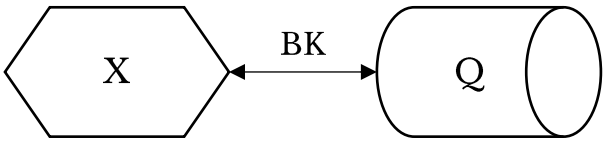
2.3.1 Exchanges and Queues private/public access

In centralized Pub-Sub protocols X and Q are virtually stored on a broker server and there are strong limits to the implementation of open infrastructure. In public blockchains X and Q are instead visible to all nodes so they are virtually public. In the basic DPS Smart Contract P have to be approved by X as well as X with Q and S with Q. Some application could instead need to open the X or the Q in order respectively to allow anyone to publish data or to receive them.

2.4 Binding Keys

Exchanges and Queues are bounded together through string keys called "Binding Keys" (BK), the binding process follows this rules:

- BK is an alphanumerical non-empty string. The use of the only non-alphanumerical character '-' is allowed;
- an X can submit a binding request to a Q specifying the associated BK;
- the Q can approve or reject the binding request;
- an X can define bindings with multiple Q;
- an X can define many bindings with the same Q;
- a Q can accept bindings from multiple X.



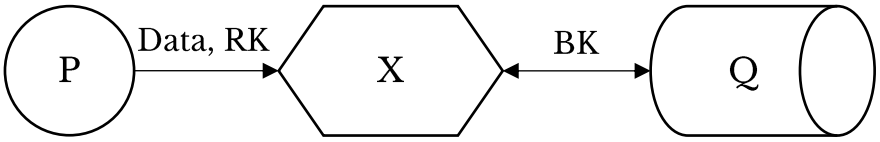
2.5 Routing process

Exchanges receive from publishers two parameters: the data that have to be sent and the "**Routing Key**" (RK). X matches the given RK with respect to all the BK stored, if it finds a match, the data will be sent to the Q bounded with that particular BK.
Matching rules rely also on regular expressions to work, with the first release of DPS Smart Contracts two regexp are implemented:

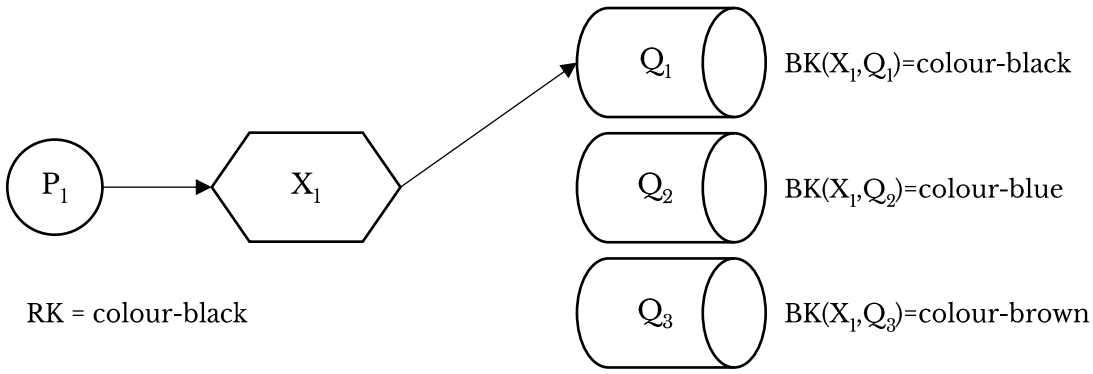
- '*': indicates 1+ valid characters;
- '#': indicates 1 valid character.

There are three main methods to deliver the message to the queues:

- **Direct**: no regular expression involved, RK and BK must be identical to match;
- **Topic**: using '*' or '#' within the RK it's possible to deliver a message to clusters of queues;
- **Fanout**: using '*' as RK every Q bounded to the X will receive the data.

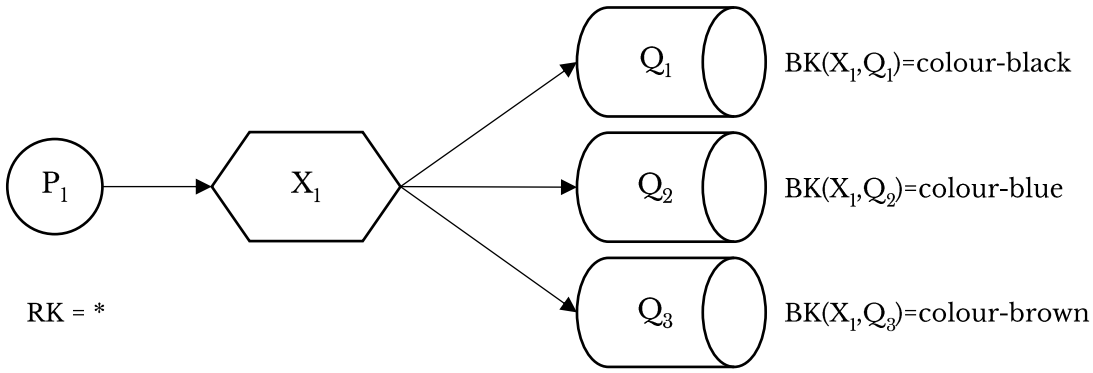


2.5.1 Direct delivery



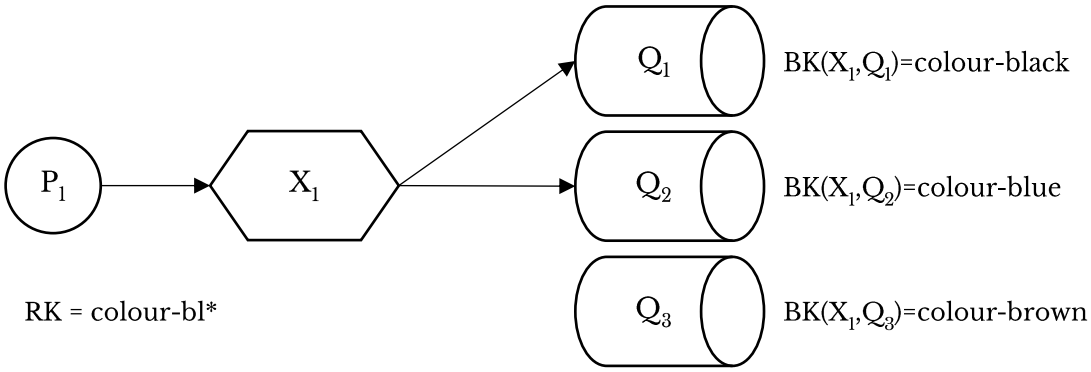
The publisher P_1 sends data to the exchange X_1 specifying "colour-black" as RK. X_1 cannot find regular expressions in RK so matches it with the stored BK to find an identical correlation, in this case $RK = BK(X_1, Q_1)$ so the message will be sent to Q_1 and then delivered by the queue to all of its subscribers.

2.5.2 Fanout delivery

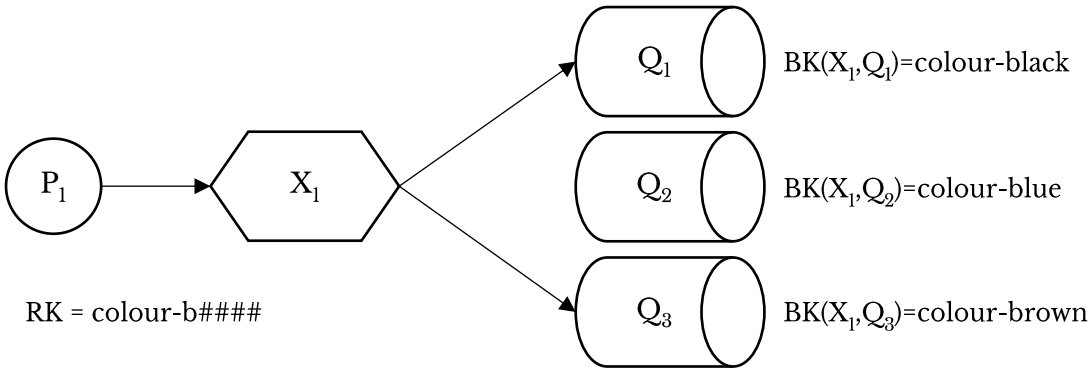


If the RK is '*' the message will be delivered to every Q bounded to X_1.

2.5.3 Topic delivery



In this case the RK contains a regular expression. '*' tells X_1 to accept an undefined number of any valid character after the root "colour-bl" in the matching with the BK. In this case the message will be sent to Q_1, Q_2.



In this case the RK contains a regular expression. '#' tells X_1 to accept a single valid character. In RK there are four '#' which means that the match with $\mathrm{BK}(\mathrm{X}_1, \mathrm{Q}_1)$ and $\mathrm{BK}(\mathrm{X}_1, \mathrm{Q}_3)$ will be succesful. $\mathrm{BK}(\mathrm{X}_1, \mathrm{Q}_2)$ does not match the regular expression because the BK has a different string length with respect to the given RK.[1]

[1] This is a rule of the current implementation and so there is the possibility to consider Q_2 as a valid bounded queue if the Exchange accepts also "no-char" as a valid value for the "#" regexp

3 Possible further developments

DPS has to be considered a contribution to the blockchain community and so it's an open and dynamic concept which could be upgraded by everyone. Here you can find some suggestions on how DPS could grow in the near future.

3.1 Release conditions for the Queues

The basic DPS Smart Contract for Queues operates in a very simple way: when it receives data by an exchange, it immediately delivers it to its subscribers. This is a very simple system which could fit needs in many cases. Different situations could need a control of the delivery instead:

- with respect to date time, environmental conditions or depending on data or events generated by third parties;
- through different approach in the release introducing LIFO method (the basic Smart Contract operates using FIFO);

3.2 Regexp engine

Current DPS implementation involves the usage of two regular expression through the two special characters '*' and '#'. The current released code is based on Solidity which has not yet implemented a regexp engine so that the two methods are directly coded in the released Smart Contracts. The advanced usage of regular expressions in DPS will be available in the future:

- Thanks to Solidity advances;
- Within other possible environments to develop Smart Contracts;
- Thanks to improvements led by the open source community.

3.3 Payments

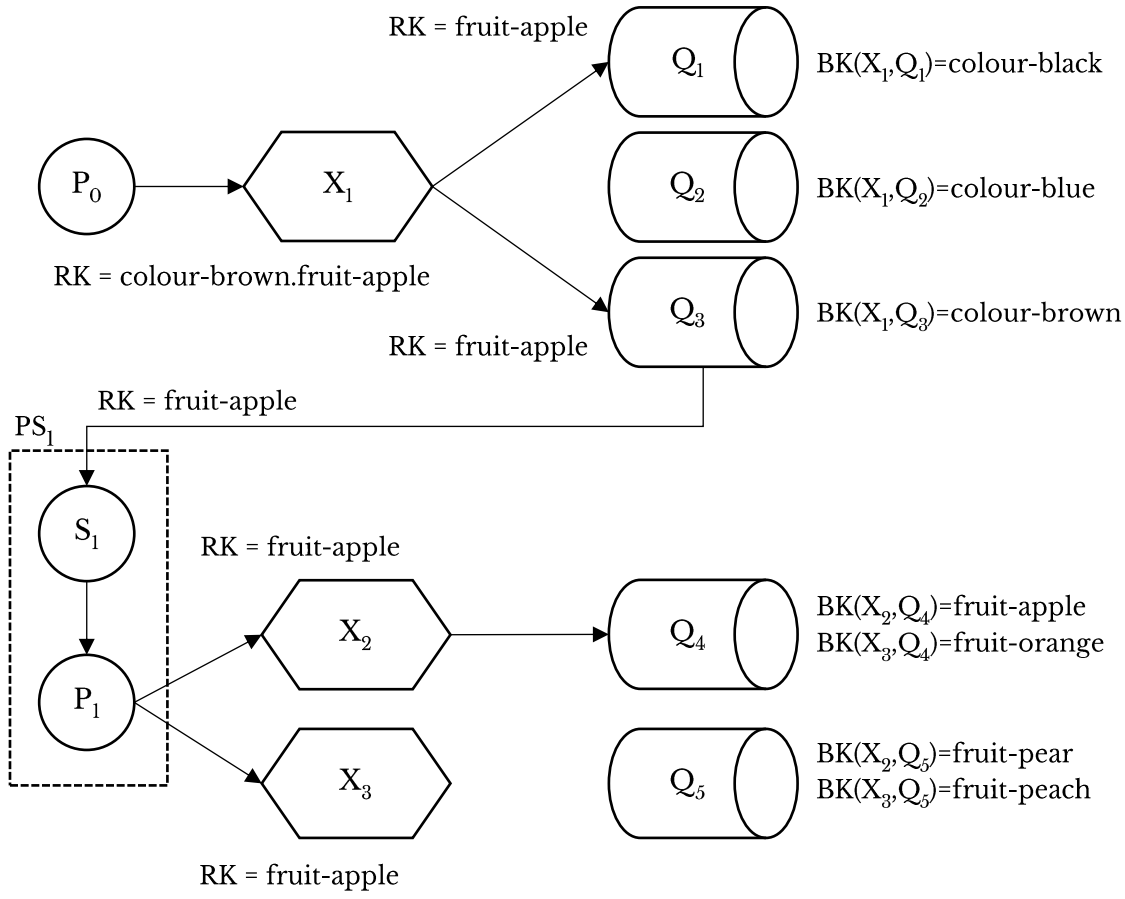
Current DPS implementation does not involve payments but its infrastructure could easily lead to include them in transaction. A publisher would be able to send value through transaction. Exchanges will receive the value and send to Queues which will have to split it to their subscribers. Following this concept Exchanges and Queues will have their own balance. The addition of release conditions to this aspect will lead to many other possibilities on the application side.

3.4 Hierarchical Routing Keys

A Hierarchical Routing Key (HRK) has the same role of a RK: it has to be specified by the P when sending data to the X to route properly the message to Q. A HRK is a sequence of different RK separated by the special char ".":

$$HRK = \{RK_1\}.\{RK_2\}.\{\dots\}.\{RK_n\}^2$$

An X receiving an HRK takes the first RK in the string to route the message and then transmit the remaining string ($\{RK_2\}.\{\dots\}.\{RK_n\}$) to the bounded Q.
A Smart Contract or a generic application on the subscriber side would recognize the pattern and then transmit data to associated X with the $HRK = \{RK_2\}.\{\dots\}.\{RK_n\}$
This concept would be useful to automate complex delivery processes and generates different topologies of distribution networks.



In this example P_1 sends to X_1 a message specifying a HRK. PS_1[3] receives the RK from the Q so it publishes the data to all bounded X attaching the given RK.

[2] Brackets indicates a placeholder so they are not valid characters for Routing Keys
[3] HRK implementation needs the use of PubSub entities. This because the subscriber which receives the RK has to be able to publish to Exchanges.

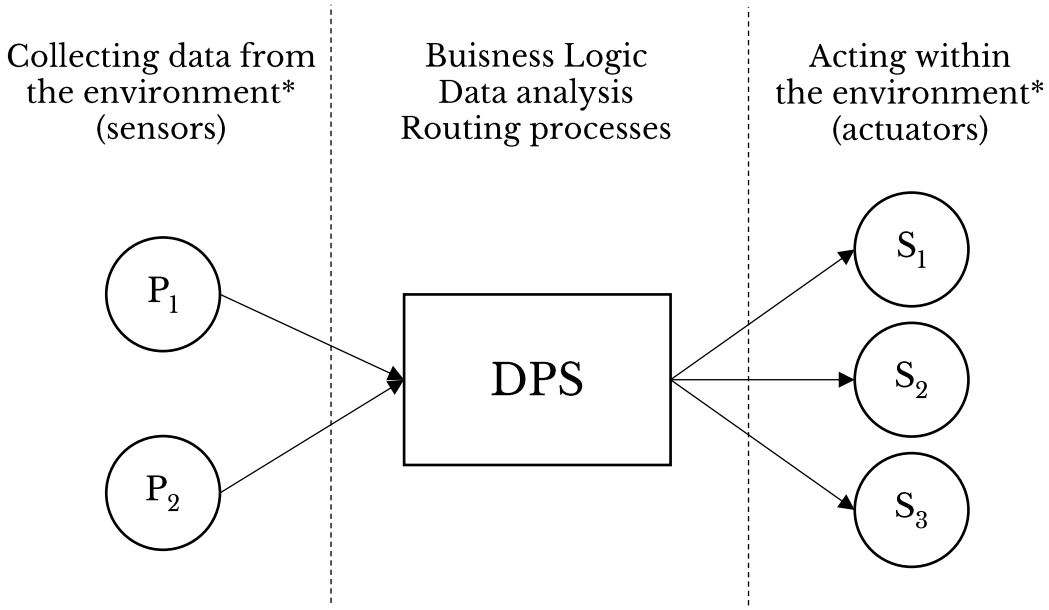
4 DPS-based applications

As any blockchain new concept or technology teaches us: there are no limits in what people can do. DPS will enable students, professionals, companies and the public administration to create innovative applications extending the limits of the basic P2P protocol. The most interesting aspect of DPS is that it translates the concept of Pub-Sub protocol in the decentralized world so it could potentially convert all of the pub-sub based centralized applications of today. Here there are some suggestions on which fields DPS could revolutionize.

4.1 Internet of Things

Today IoT is one of the most active area in centralized pub-sub protocols adoption because of its nature: sensors have to be considered as publishers and actuators as subscribers. The implementation of DPS in IoT could make a step forward in the automation of networks of smart devices, in the interaction within the networks and between different networks. Main examples of areas interested by DPS are:

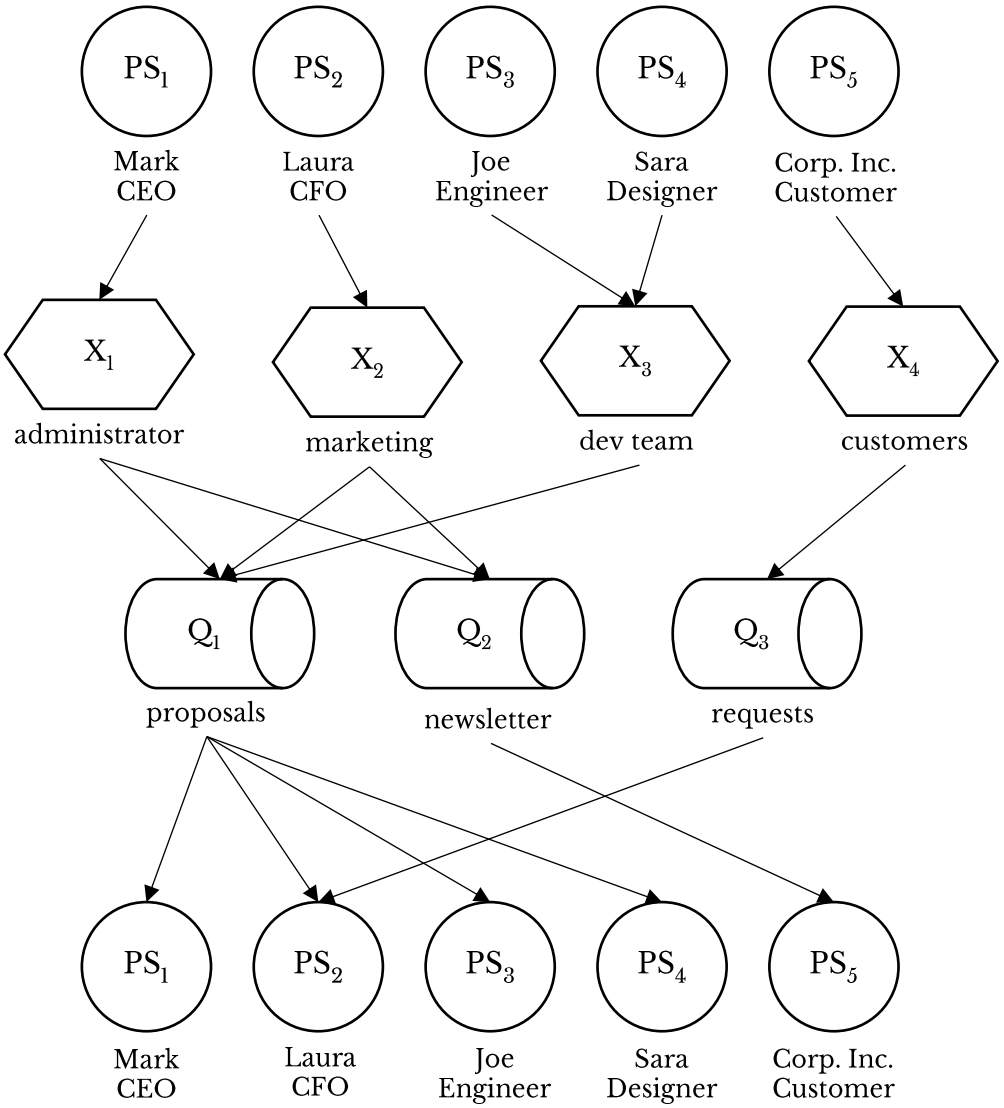
- domotics and building automation;
- smart cities;
- autonomous vehicles;
- industrial processes and machines;
- agricolture.



*Or through third parties

4.2 Democracy and organizations

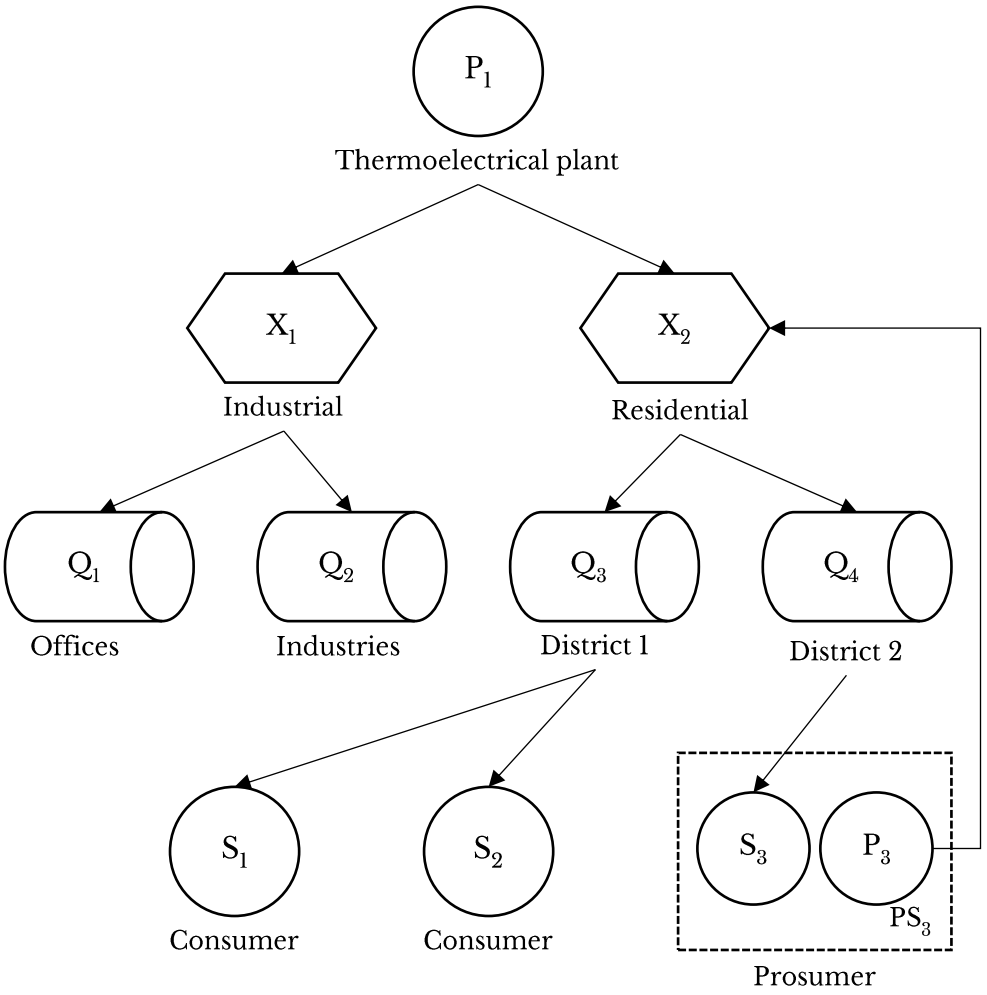
The introduction of the DAO (Decentralized Autonomous Organization) paradigm, enforced by the pillars of blockchain (anonymous transactions, public ledgers, system security) is one of the most important advances in the decentralized world because it regulates any possible organization: from the small community to the corporate environment, from the fundraising syndacation to a government. DAO relies on basic peer-to-peer but a DPS integration will lead to new horizons, in particular in making the members interacting with each other and in managing the execution of a proposal regarding a large set of subscribers.



In this example DAO and DPS are intended to be integrated to introduce a system based on proposals in a company. The example in addition shows how easy could be the integration with third parties, for example with a software to engage customers which will be Pubs (when they send requests) and Subs (when they receive the newsletter).

4.3 Energy, water and natural gas distribution

It's a well known and accepted concept the one who predicts that in future electric smart grids will be based on blockchain technology. Extending this thought to other aspects such as natural gas or water distribution it's easy to think about Smart Contracts acting as counters and controllers of the incoming/outcoming flows. DPS could revolutionize this world introducing a mechanism which really fits the needs of modern concept in energy where the *prosumer*[4] word indicates a node of the network capable of both consuming and generating energy (through PV, hydro, wind, …).



In this example the prosumer in District 2 consumes the energy given by the network but it's also capable of generating it, being so a publisher in the system which gives energy to the network itself.
DPS will allow powerful smartgrids to be created thanks to an efficient organization of the network.

[4] The concept of prosumer is easily implemented as a Smart Contract after inheriting the methods of the Producer (Publisher) and the Consumer (Subscriber) Smart Contracts.

4.4 Medical, healthcare

Data on people health are one of the most strictly regulated information in the world so that many companies are already developing blockchain applications to solve the security problem of such situations. The introduction of DPS, integrated with digital identity systems, will help hospitals to keep track of their patients. Smart Contracts will interact with other ones on wearable devices and immediately send information to the subscribers (for instance patient's relatives and doctor).

4.5 Supply chain, logistics, shipments

The incomparable level of security introduced by blockchain technology really fits in the product/process chain ecosystem. In food is fundamental to know the origins of ingredients and to keep track of the movements that the dish made all around the world. In a near future this will be fully automated thanks to Smart Contracts operating on various levels on the supply chain which involves different players on every stage: professionals, companies and regulators that should start "publishing" every information.
The complete chain of transaction will be on the public ledger so it will be available for the consumer.
This will refer not only to food but to every product and will help companies and consumers in tracking their assets.

4.6 Messaging, advertisement, people engagement

Today most of chats are based on pub-sub protocols, this also because it's natural to think about "publisher" and "subscriber" when programming an application like that. DPS will serve as the infrastructure for decentralized chats, not only peer-to-peer but adopting the one-to-many and the many-to-many paradigms.
A one-to-many application which could easily be built on DPS is a newsletter system and based on this principle you could imagine many other content-delivery applications: video streaming, couponing & loyalties initiatives, public administration announcements, gaming.

5 Conclusions

In the end of this paper author's hope is to have been able to involve passionates around a topic which could represent a relevant step forward in blockchain technology adoption, expecially on the professional side.
DPS has the advantage of representing the decentralized version of a well known and widely adopted paradigm in the technology today so it would be easy to convert nowadays centralized applications in blockchain ones.
In the near future DPS will be improved as a public asset for blockchain by the community grown around and than the whitepaper also will be released in further versions.