**Functionalities and features:**

Am plecat de la un articol in care este descris un mecanism criptografic hibrid, pe 2 nivele:

* Primul nivel cu DES
* Al doilea nivel cu RSA

Mecanismul criptografic va fi folosit in dezvoltarea unui proiect pentru asigurarea securitatii datelor in cadrul unui smart house.

Consideram contextul proiectului urmatorul: un smart house care contine un server central si mai multe sisteme embedded folosite in alte scopuri. Proiectul va fi dezvoltat urmand paradigma master slave.

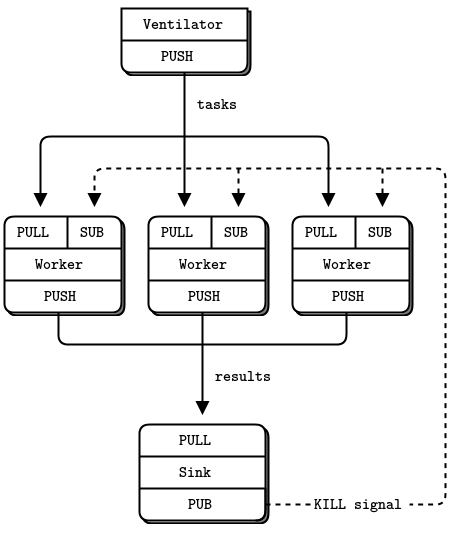
Datele sunt impartite worker-ilor pentru procesarea criptografica, iar apoi colectate de server-ul central pentru stocare. Vom urmari si elasticitatea algoritmului in functie de distributia pe tipuri de worker.

Pentru asigurarea confidentialitatile datelor vom folosi algoritmul criptografic descris. Iar pentru asigurarea integritatii datelor vom folosi functii hash.

**Technologies used:**

Proiectul va fi scris in limbajul c++, utilizand librariile criptografice openssl si libraria Openmpi.

Openmpi asigura o comunicarea asincrona intre server si workeri.



Server-ul actioneaza ca un ventilator de task-uri. Workerii disponibili fac „PULL” catre server, iar server-ul distribuie task-uri. Rezultatele sunt transmise server-ului printr-un „PUSH”. Server-ul foloseste un socket „PUBLISH” pentru a publica un semnal „KILL” in cazul in care este nevoie oprirea procesarii.

**Use cases:**

1. Generarea datelor pe master
2. Cautarea worker-ilor disponibili pentru procesare prin publicarea unui semnal de procesare catre workeri.
3. Workerii fac „PULL” catre master
4. Master-ul trimite task-uri de procesare
5. Procesarea criptografica pe workeri
6. Workerii fac „PUSH” catre master cu rezultatul obtinut
7. Colectarea rezultatului pe master

**Integrarea cu yocto este optionala.**