

A Digital Twin-based Smart Home: A Proof-of-Concept Study

Student: Laura Corssac - Supervisor: Prof. Dr. Juliano Wickboldt - 10/05/22



O meu trabalho é uma prova de conceito de uma casa inteligente baseada no conceito de gêmeo digital

Summary

1. **Introduction**
2. Use Case 1
 1. Introduction
 2. Architecture
 3. Demo
3. Use Case 2
 1. Introduction
 2. Architecture
 3. Evaluation
4. Limitations
5. Conclusion

Vou iniciar contextualizando o assunto

Digital Twins



Gemeos digitais são replicas digitais de objetos físicos.

Essas cópias constantemente recebem dados de sua contraparte real através de sensores, de fontes externas e de usuários.

Digital Twins



Digital Twins



Digital Twins



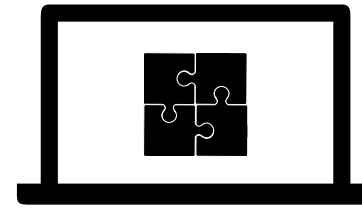
Digital Twins



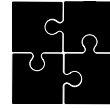
Com tudo isso a replica consegue simular comportamentos do objeto físico, fazer predicoes e tomar algumas decisoes que podem ser devolvidas a ele.

Também consegue prover ao usuário análises do objeto e uma descrição completa dele.

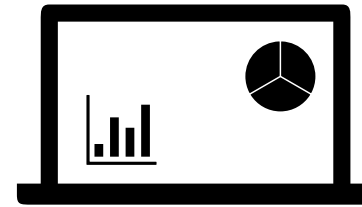
Digital Twins



Digital Twins



Digital Twins



Digital Twins



Domotics, Home Automation



5

A área de domótica, ou automação residencial, visa a automatizar algumas ações em uma casa, tornando a uma casa inteligente. Por exemplo, podemos controlar a intensidade de uma luz com base na iluminação ambiente.

Domotics, Home Automation



Proposal



6

Apesar de os dois conceitos serem muito populares não há muitos projetos que unam os dois. Então esta é a nossa proposta: provar que tecnologias de DT também podem beneficiar residencias, tornando-as inteligentes.

Um DT de uma casa tem funcoes mais amplas que estrategias tradicionais de casas inteligentes. Ele não só produz automacoes como também análises e simulações. Além disso, busca uma maior precisao nos resultados, combinando dados do sistema físico com os de uma modelagem digital de uma casa.

Proposal



Digital Twin Roles

7

A criação do termo do DT é atribuída ao cientista Dr Michael Grieves e ele divide o papeis dos DT em 2 sub-categorias.

clica

A primeira é a interrogaritva se resume em permitir que ele interroque o sistema a respeito de características passadas e presentes do sistema fisico.

clica

A segunda é a preditiva se trata da capacidade do DT em realizar previsões sobre o estado futuro do do objeto de estudo.

Nesse trabalho a gente apresenta um caso de uso para casa um desses papeis.

Digital Twin Roles

Interrogative

Digital Twin Roles

Interrogative

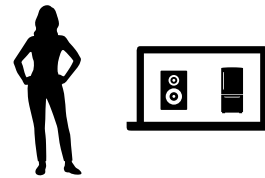
Predictive

Summary

1. Introduction
2. Use Case 1
 1. Introduction
 2. Architecture
 3. Demo
3. Use Case 2
 1. Introduction
 2. Architecture
 3. Evaluation
4. Limitations
5. Conclusion

Use Case 1

Interrogative Role



9

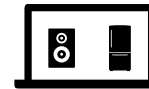
O primeiro caso de uso cumpre o papel interrogativo do DT. Ele permite que o usuário visualize o consumo de energia dos eletrodomésticos de sua casa em tempo real. Além disso, possibilita que ele consulte a média do consumo em um certo período de tempo passado.

clica

Assim, o usuário tem maiores informações sobre a atividade energética de sua casa podendo utilizá-las para racionamento de energia e melhor entendimento de cobranças financeiras.

Use Case 1

Interrogative Role



O primeiro caso de uso cumpre o papel interrogativo do DT. Ele permite o usuário a visualizar o consumo de energia dos eletrodomésticos de sua casa em tempo real. Além disso, possibilita que ele consulte a média do consumo em um certo período de tempo. Assim, o usuário tem maiores informações sobre a atividade energética de sua casa podendo utilizá-las para racionamento de energia e melhor entendimento de cobranças financeiras.

Use Case 1

Interrogative Role



01/05/22 - 10/05/22



Use Case 1



A visualização é feita através da coloração codificada por cores

clica

das representações 3D dos equipamentos eletrônicos. Assim, quanto maior for a intensidade da cor mostrada na visualização, mais energia aquele equipamento está/ ou esteve consumindo.

Use Case 1



A visualização é feita através da coloração codificada por cores das representações 3D dos equipamentos eletrônicos. Assim, quanto maior for a intensidade da cor mostrada na visualização, mais energia aquele equipamento está/ ou esteve consumindo.

Use Case 1



Summary

1. Introduction
2. **Use Case 1**
 1. Introduction
 2. **Architecture**
 3. Demo
3. Use Case 2
 1. Introduction
 2. Architecture
 3. Evaluation
4. Limitations
5. Conclusion

Como implementamos isso

O primeiro componente de nossa arquitetura é o sistema físico.

clica
Ele idealmente é composto por uma casa com eletrodomesticos com sensores de potencia em suas tomadas.

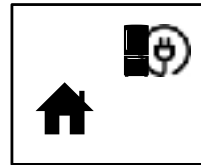
Isso não foi possível para a gente, então simulamos o espaço físico com um conjunto de dados já coletado.

Clica
Ele se trata de um CSV em que as colunas representam um conjunto de eletrodomesticos e as linhas sao suas medicoes de potencia, espacadas por 1 segundo.

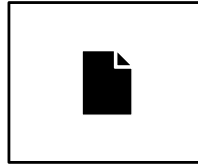
Physical System



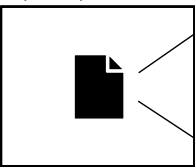
Physical System



Physical System



Physical System



Time	Fridge	Oven
1s	0.3kw	0.2kW
2s	0.5kW	0.2kW
3s	0.3kW	0,5kW

Physical System

Time	Fridge	Oven
1s	0.3kw	0.2kW

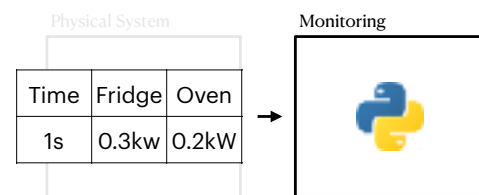
Para a então coleta de dados, fizemos um programa em python que le esse arquivo.

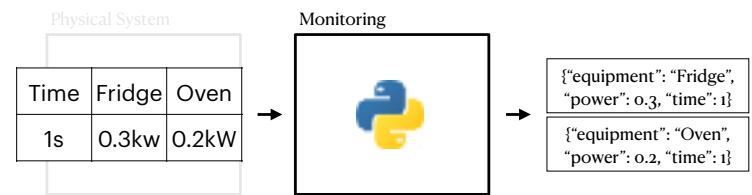
Pra cada linha,

Clica

cria estrutura de dados pra cada equipamento contendo o seu nome e a medida de potencia e o instante de tempo da linha

Clica

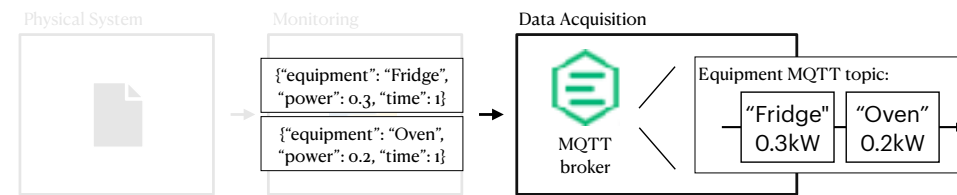


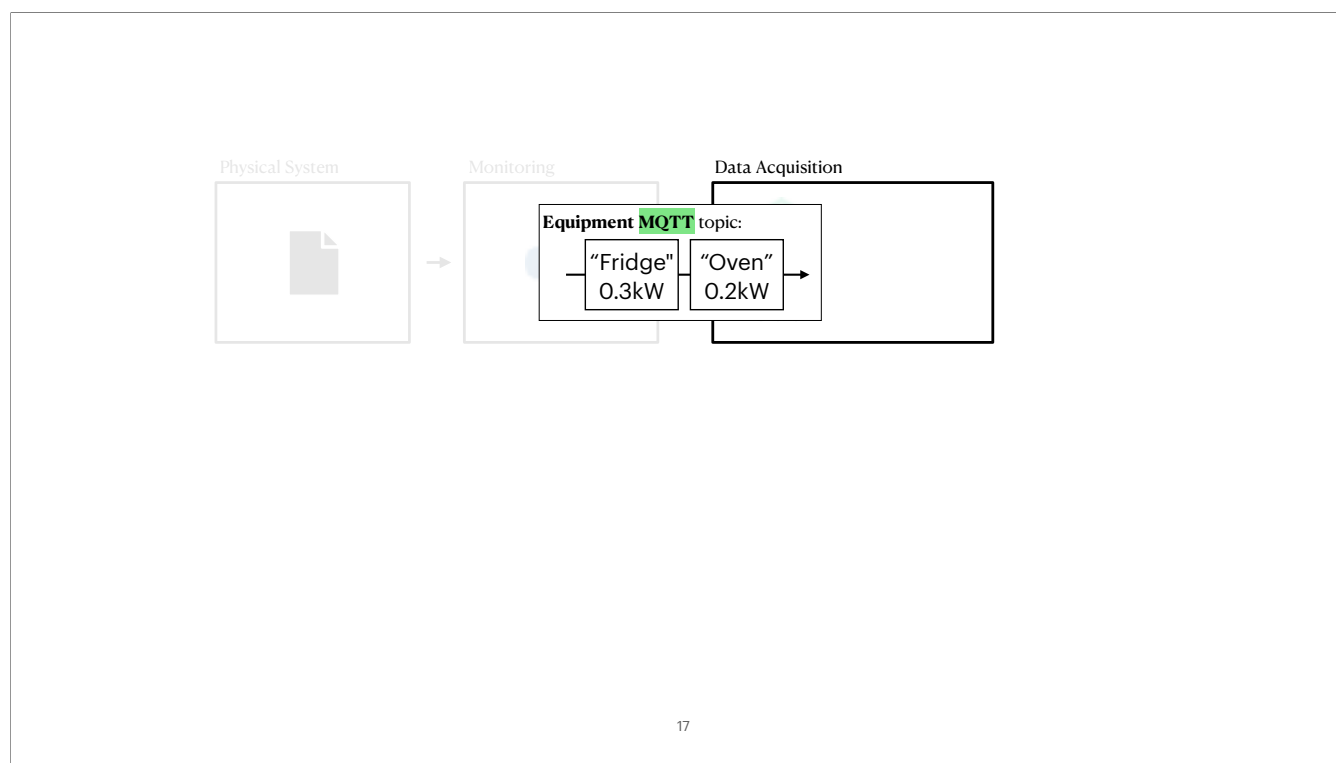




E publica essas mensagens em um tópico MQTT, que é um protocolo leve e é muito utilizado na captacao de dados de sensores, já esses são equipamentos de baixa potencia.

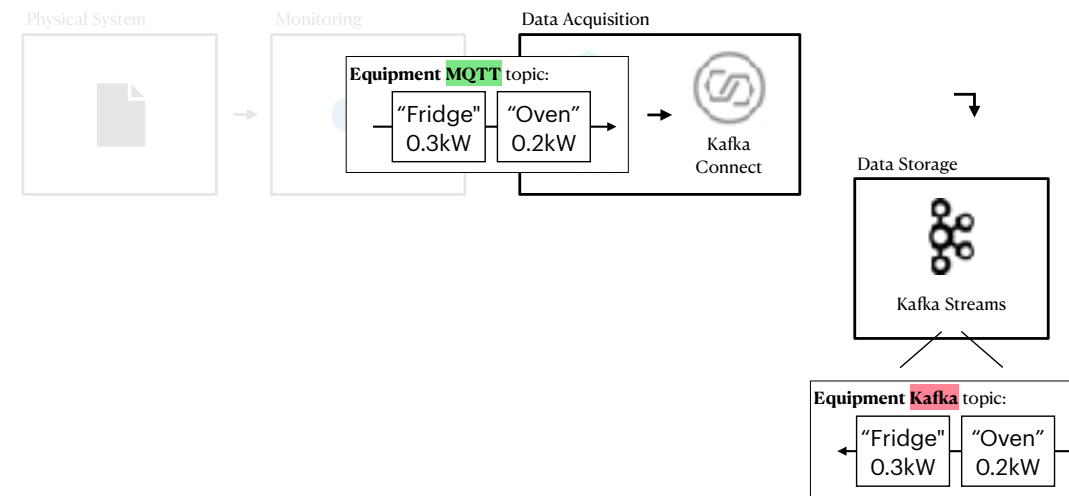
O broker utilizado foi o EMQX, cujo símbolo é esse verde.

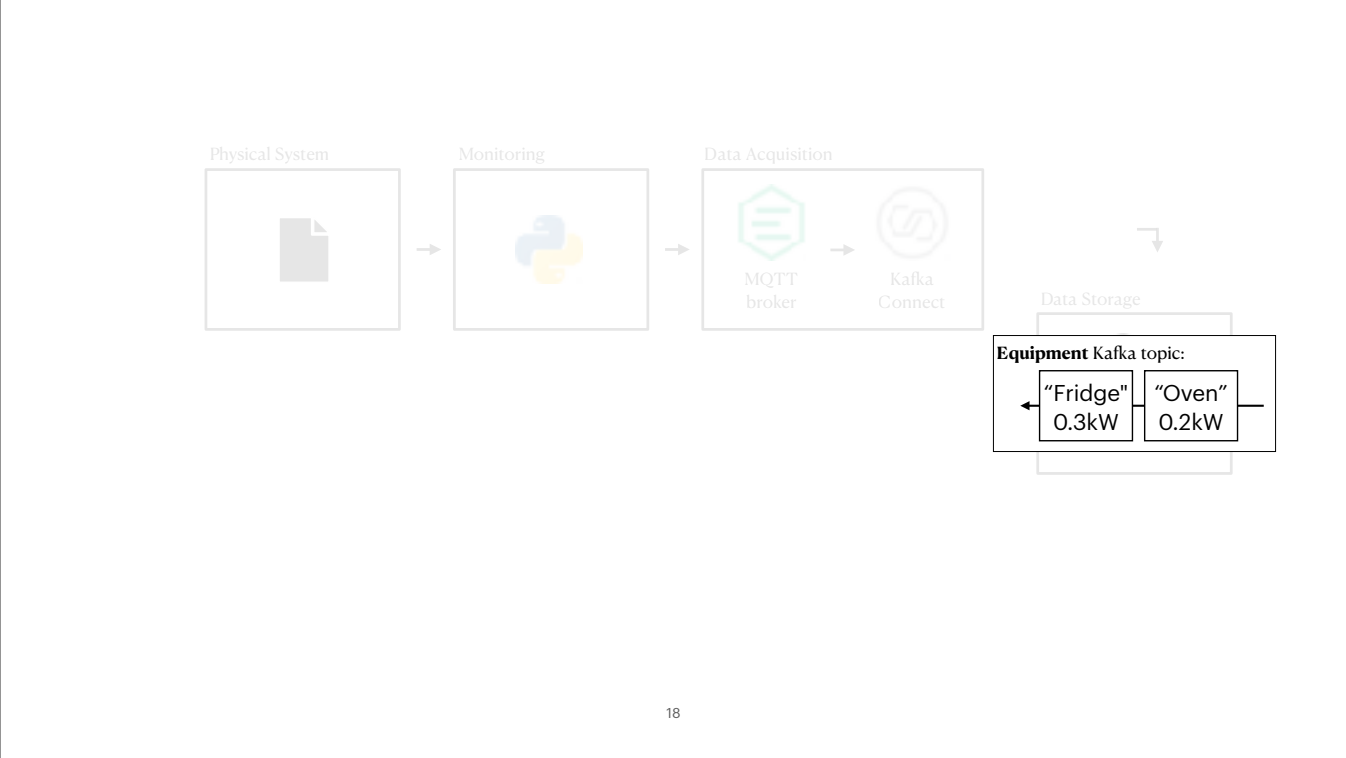




CONNECT

Esses dados então são armazenados em um Stream na plataforma Kafka. Para fazer essa transposição das mensagens do tópico MQTT para o stream no Kafka, temos um componente entre eles chamado de Kafka Connect. O que ele faz é escutar o tópico MQTT e publicar cada mensagem em um tópico Kafka, mantendo a estrutura dos itens.



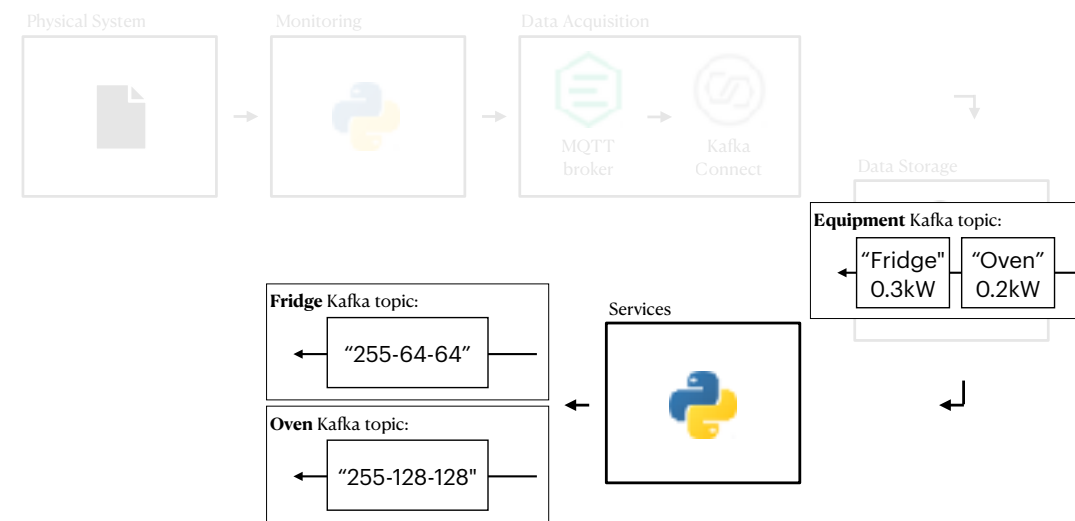


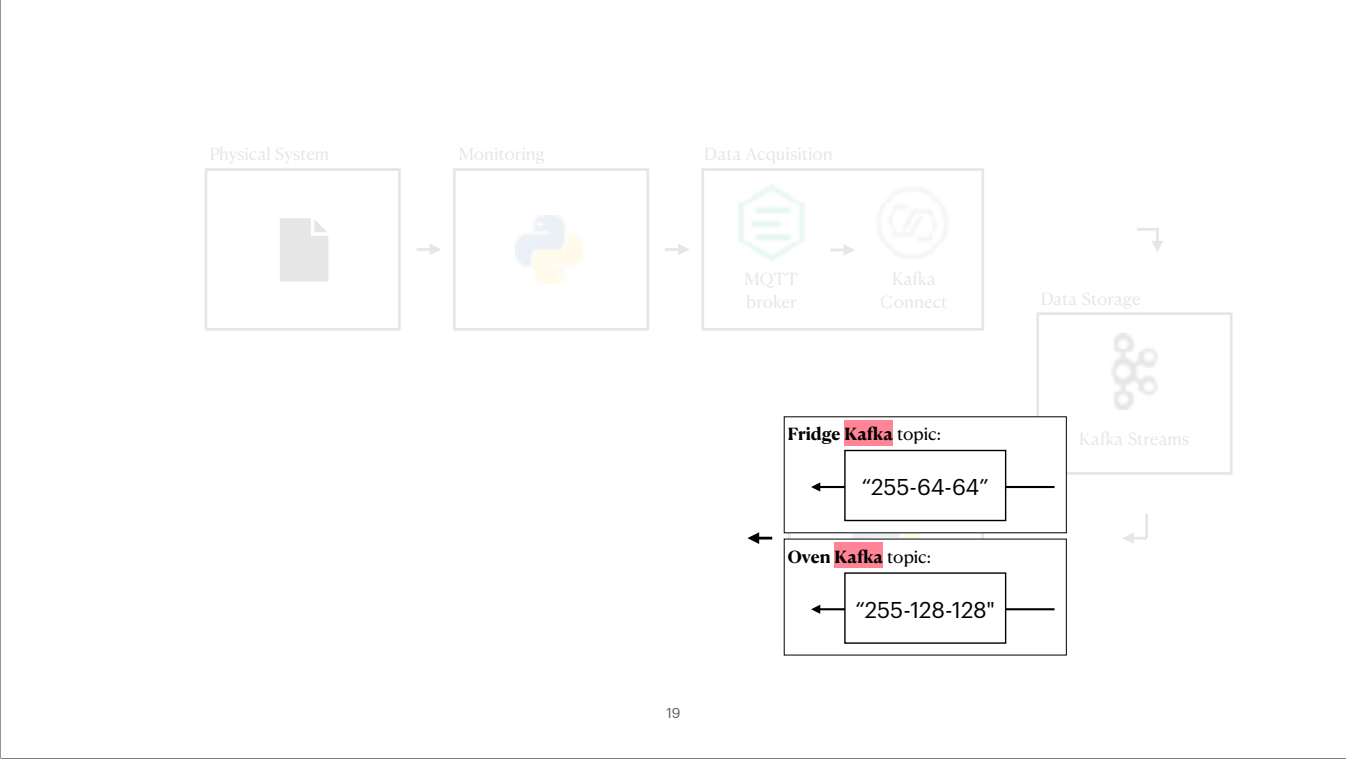
servico

O próximo componente é um serviço que vai receber os eventos esse tópico do Kafka transformar os valores de potencia para cores. Ele publica então o RGB da cor resultante, representada por uma string, em um topicos Kafka respectivo a cada equipamento.

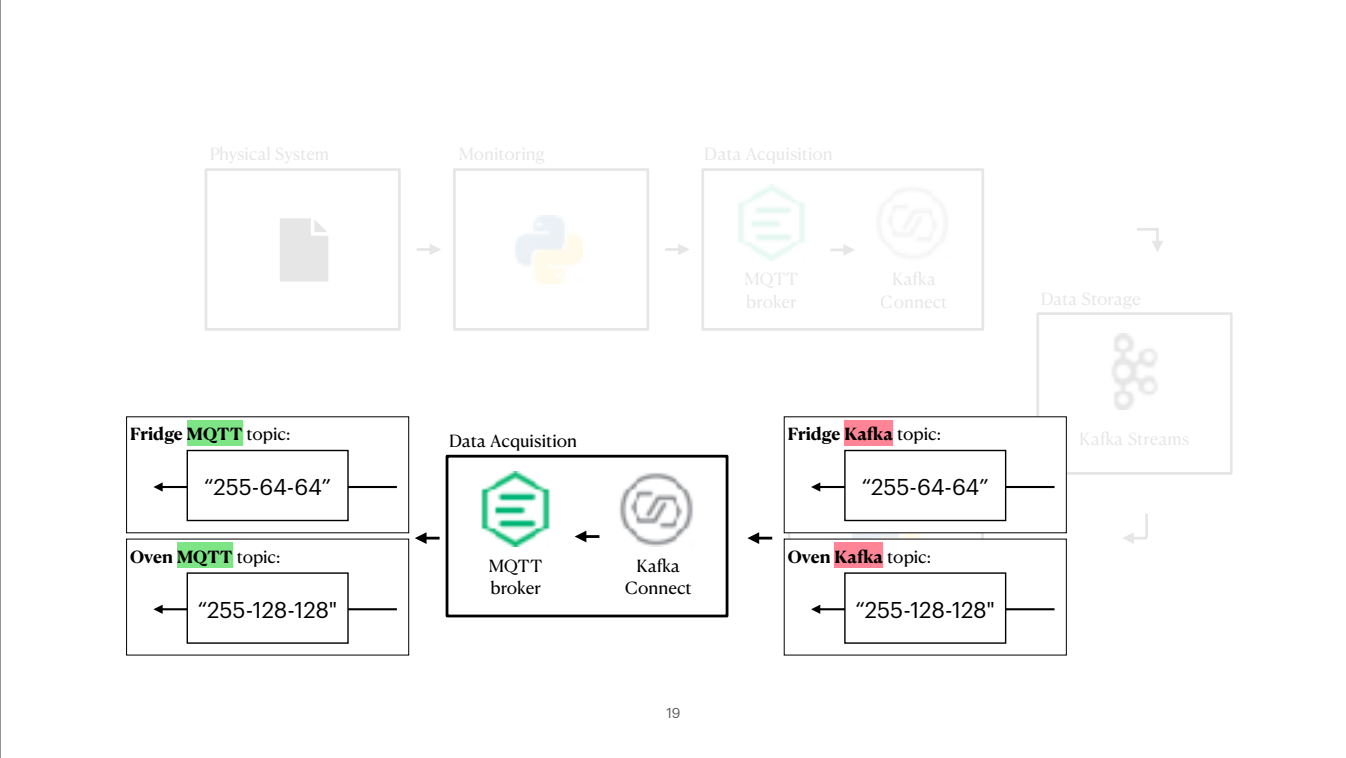
clica

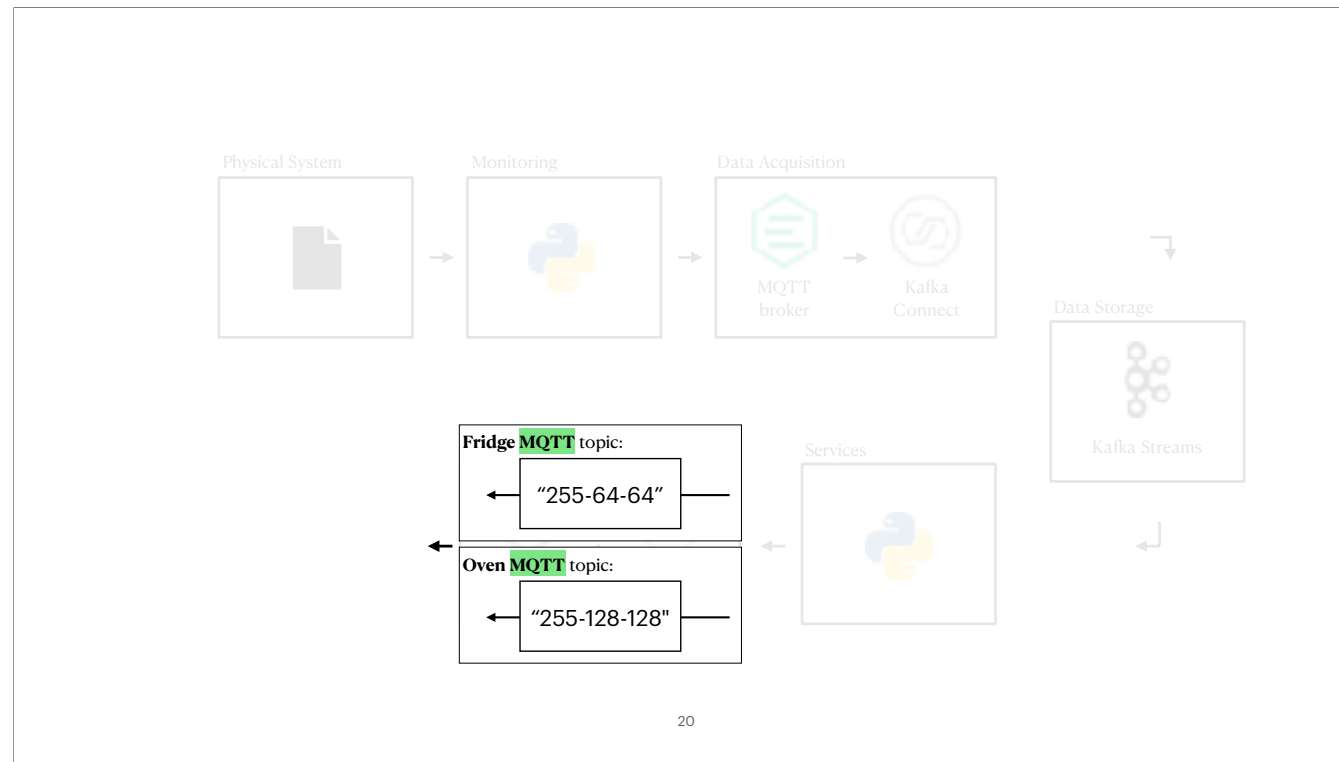
Enquanto tempos um topico de entrada, que recebe os valores dos sensores a cada segundo, temos um tópico de “saída” para cada equipamento, que recebe os dados de cores.





CONNECT - BROKER 2



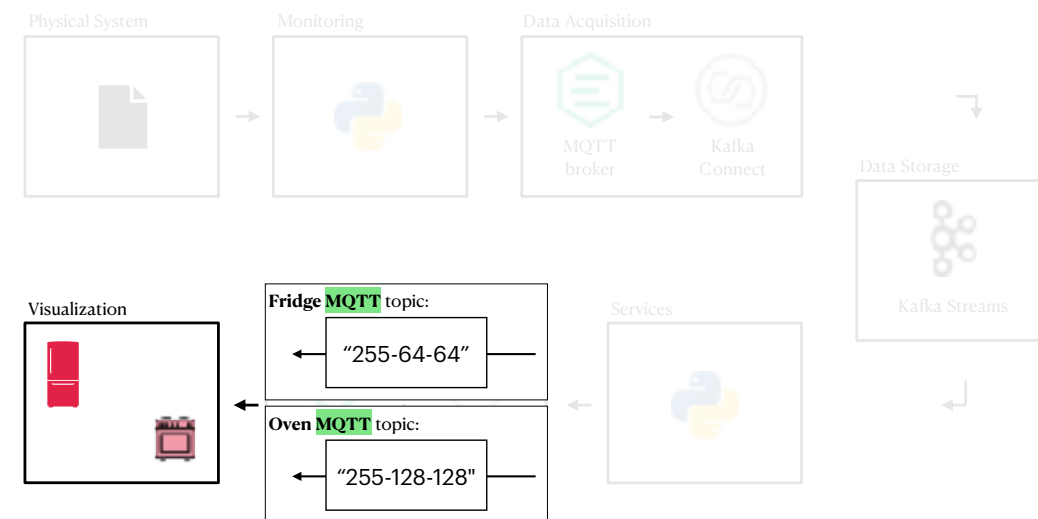


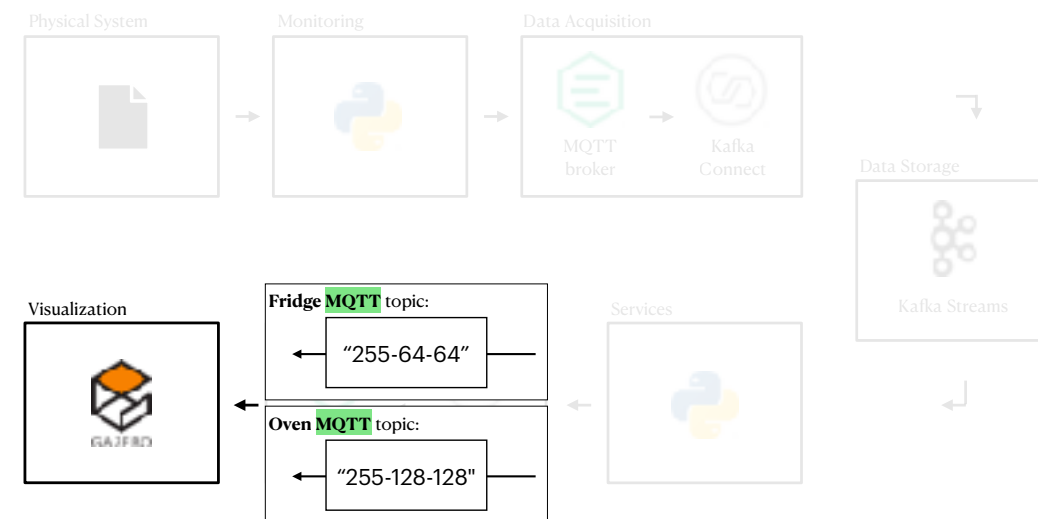
GAZEBO

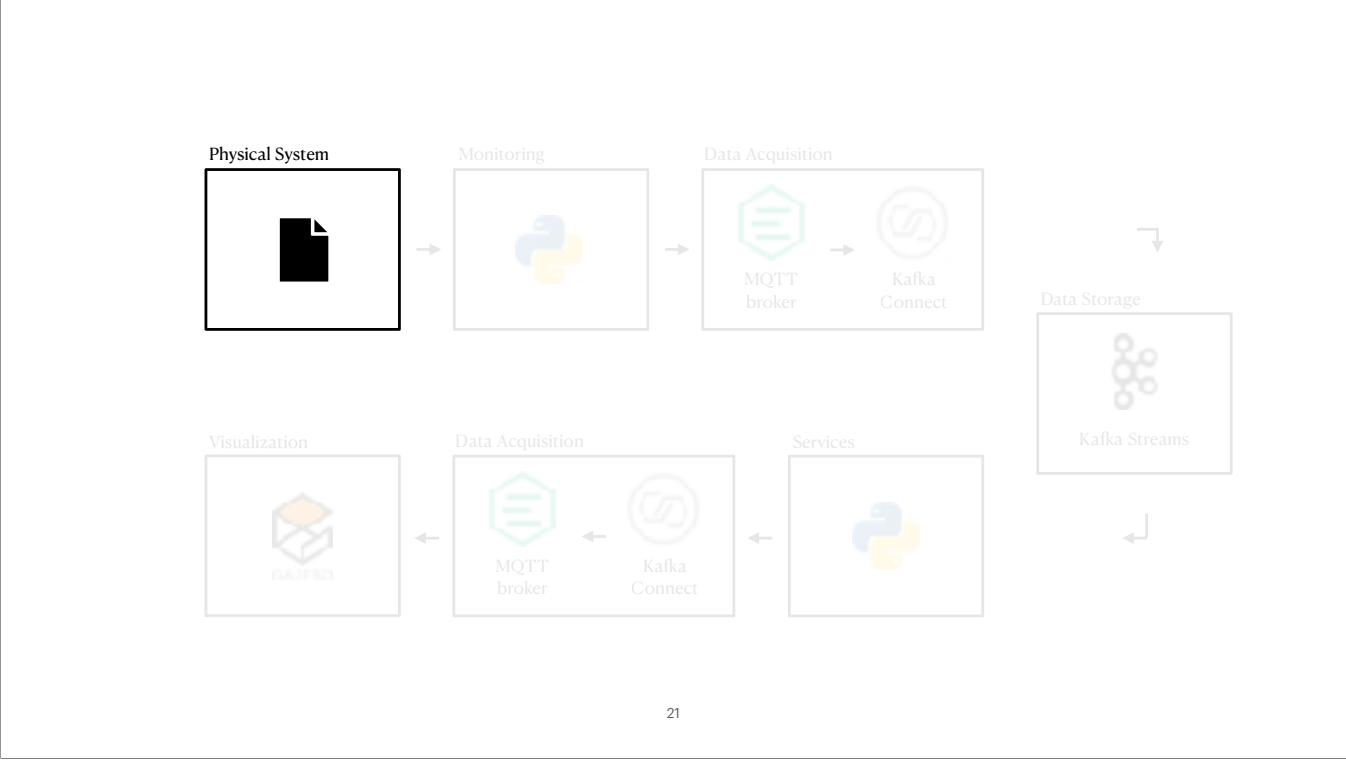
Essa informação de cor é então usada para colorir as representacoes 3D dos equipamentos da casa. Para isso, usamos um software gazebo.

CLICA

que nos permite criar um mundo em 3d, com um modelo 3D pra cada equipamento e controlá-los em tempo real. Cada objeto tem um script de controle que escuta o tópico MQTT respectivo àquele objeto.

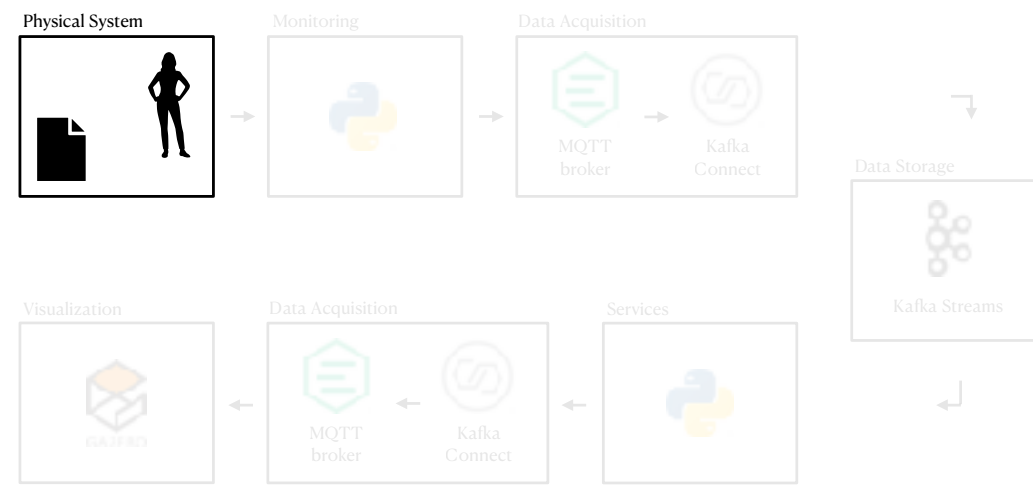


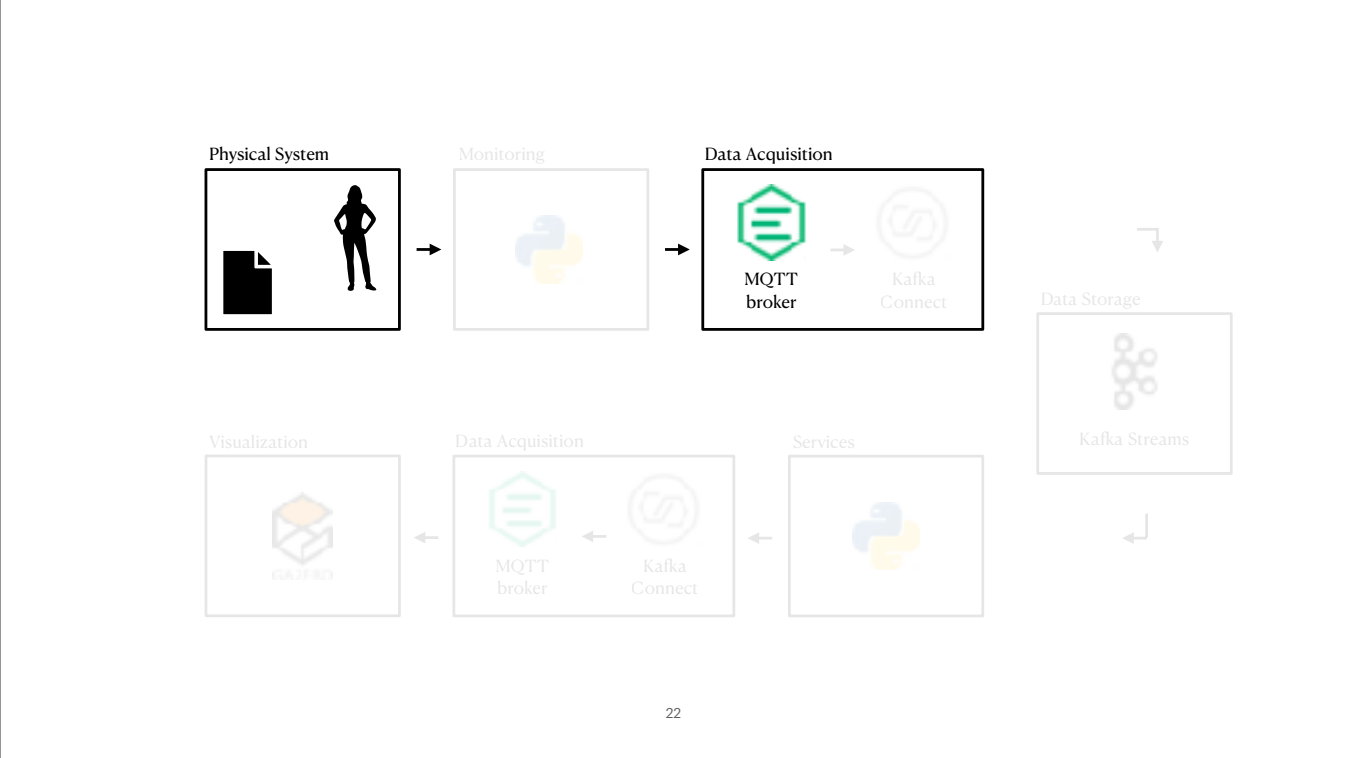




USER
Pra permitir que o usuário

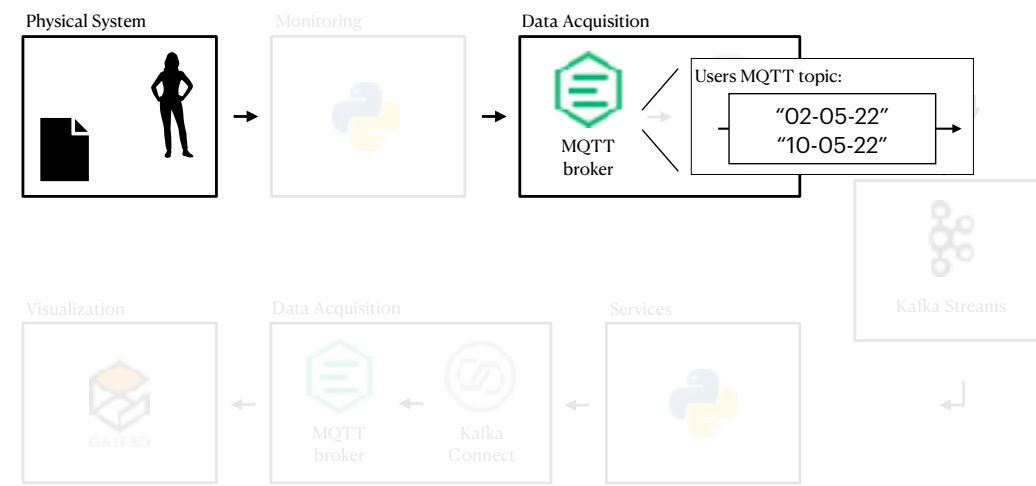
Clica
visualize a média dos valores de potência medidos em algum período,

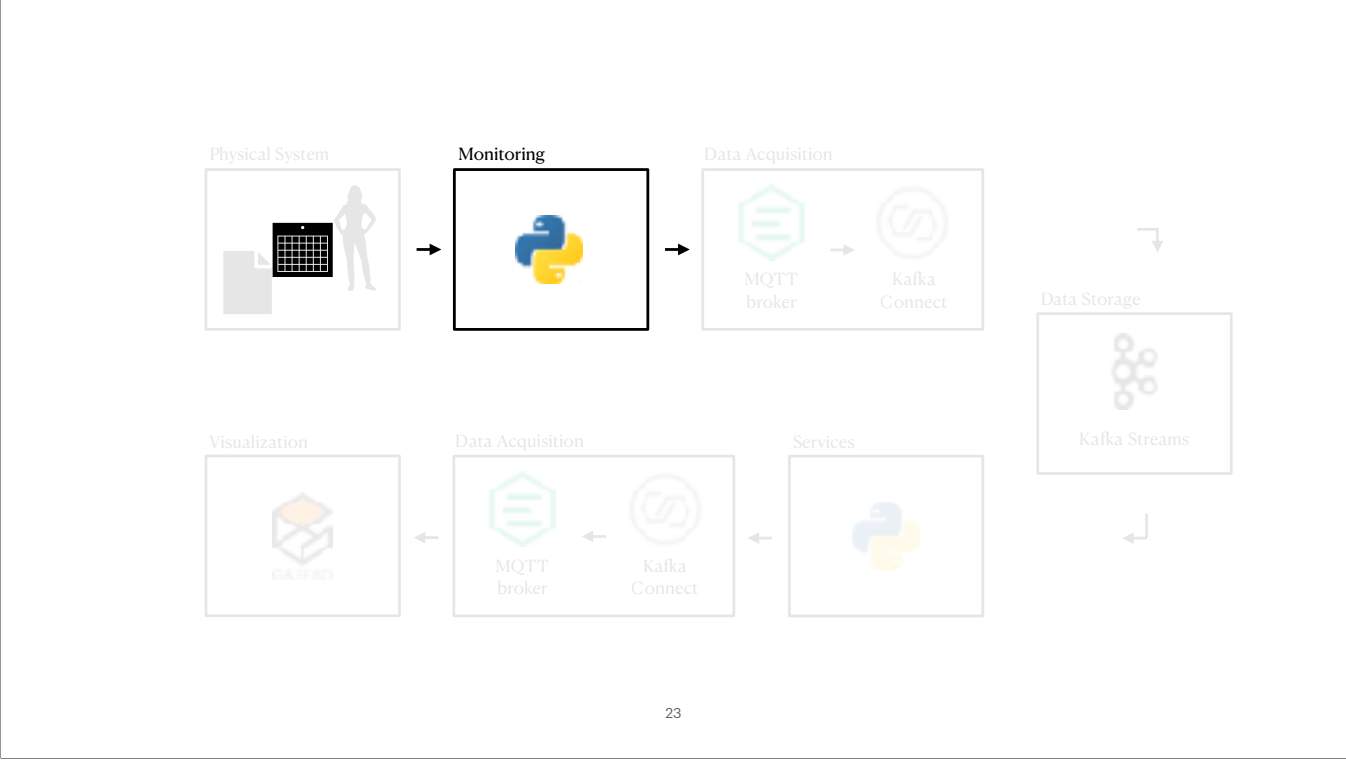




USER MQTT

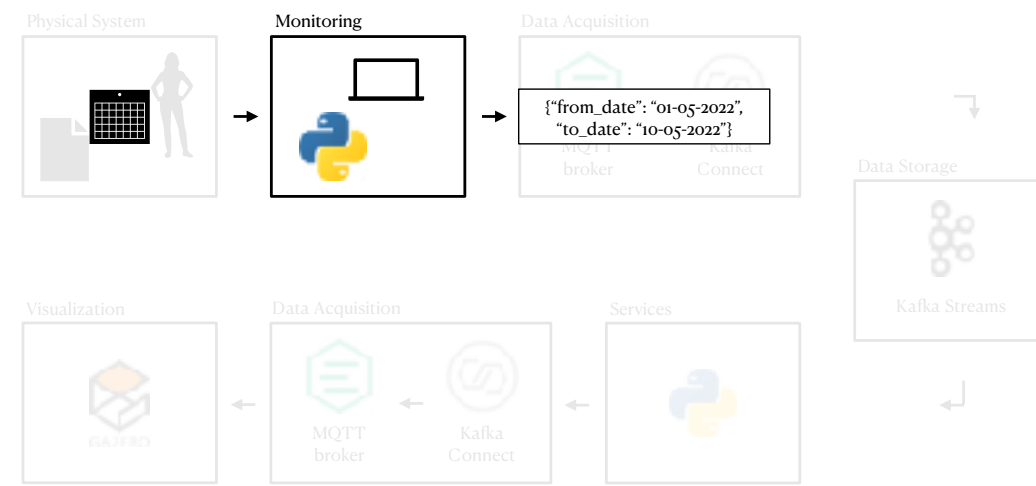
cria um novo tópico MQTT que vai receber as vontades do usuário, como a data de início e a data final do período que ele quer analisar

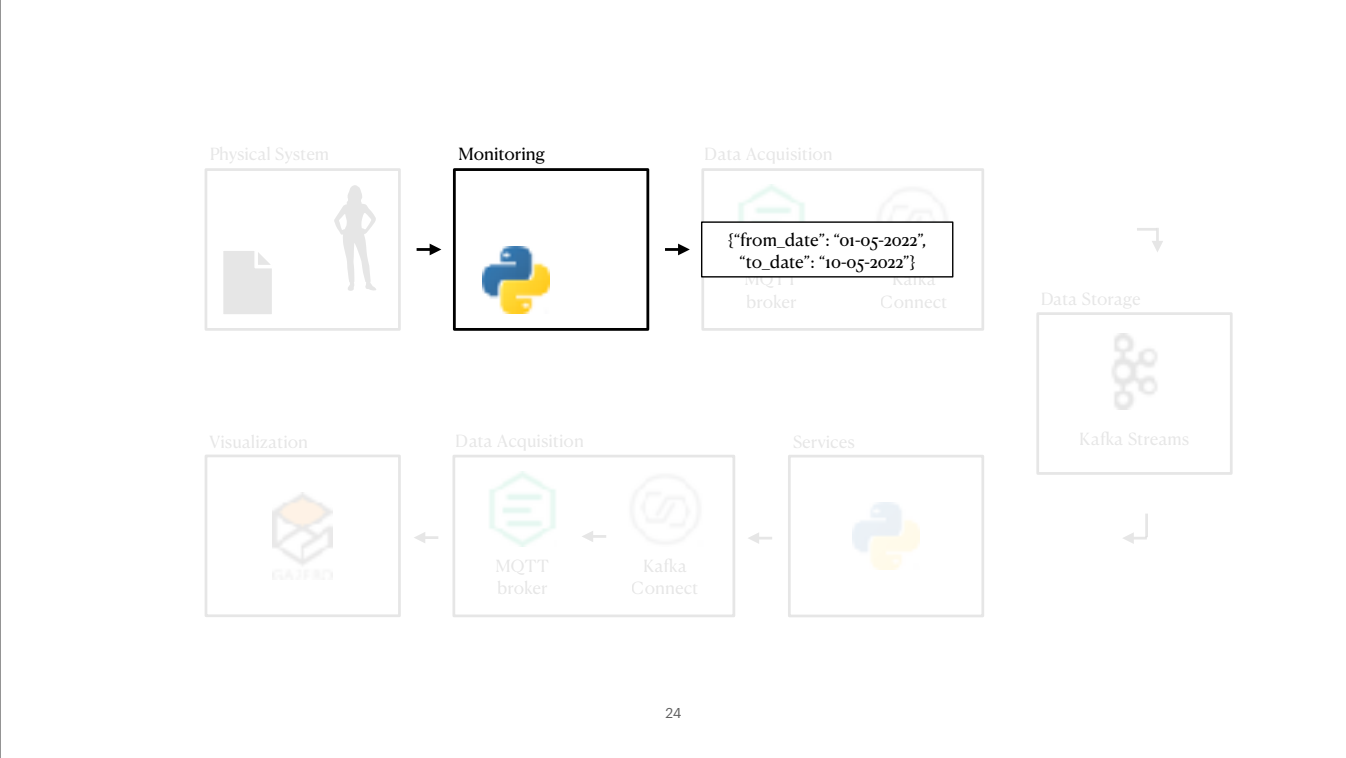




USER MONITORING FAKE

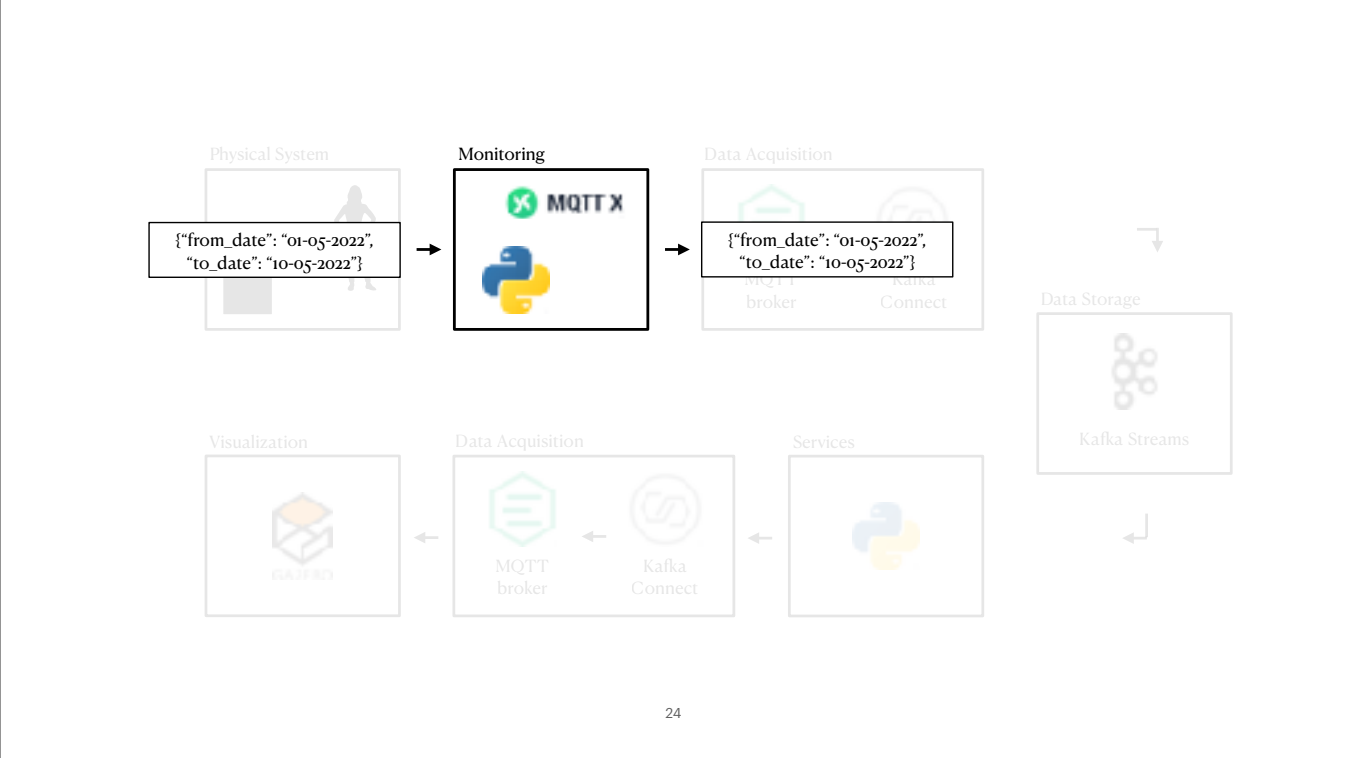
Idealmente a gente teria uma interface web, por exemplo, que disporia elementos user-friendly e traduziria entao as entradas do usuário pro formato esperado no tópico que é um JSON.





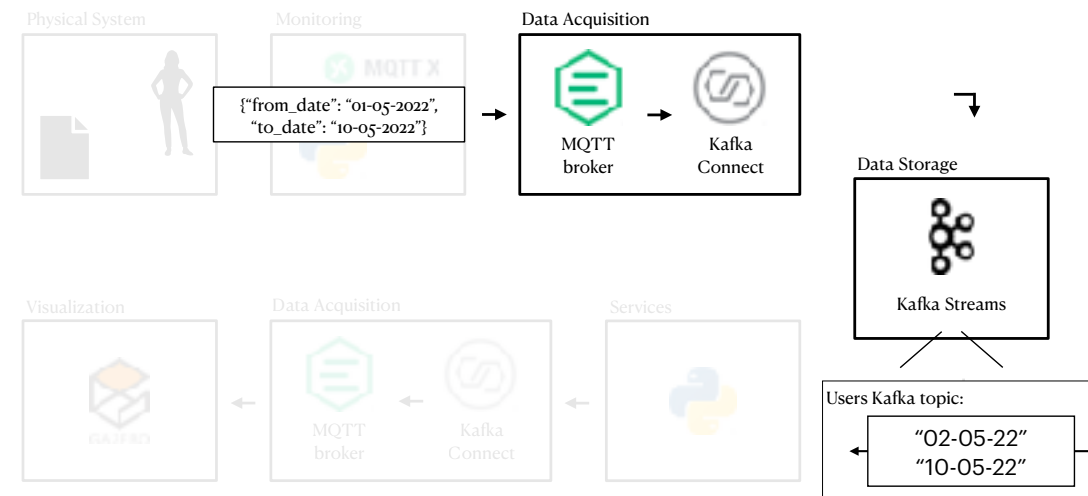
USER MONITORING REAL

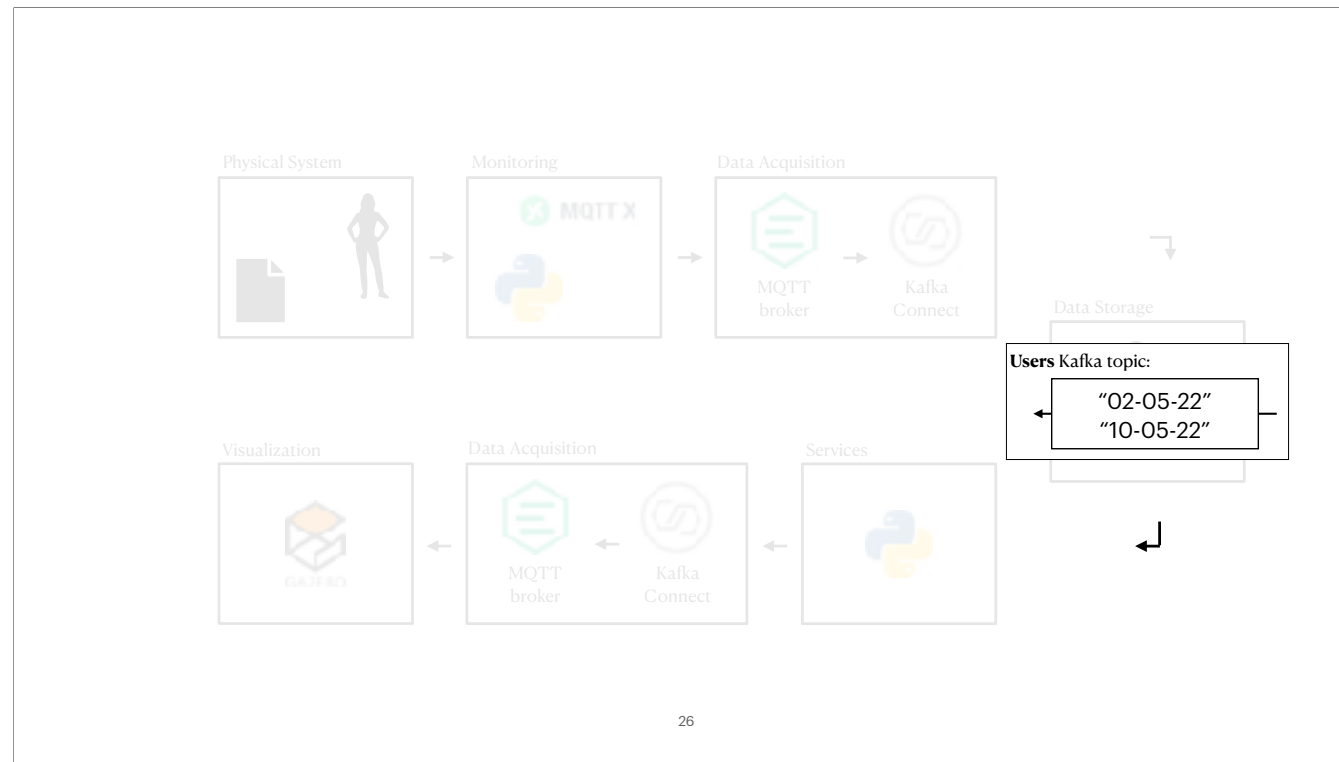
Isso nao entrou no escopo entao..





Transicao ao kafka



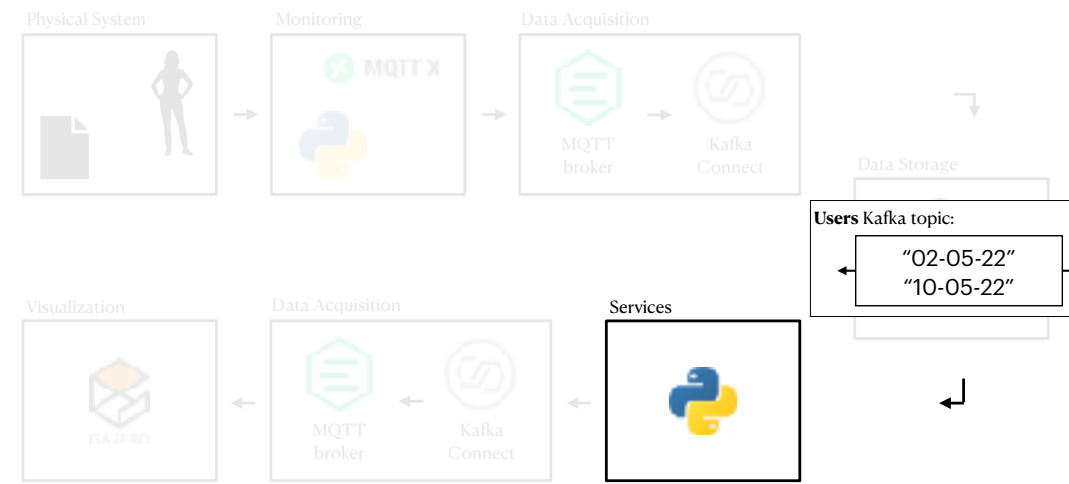


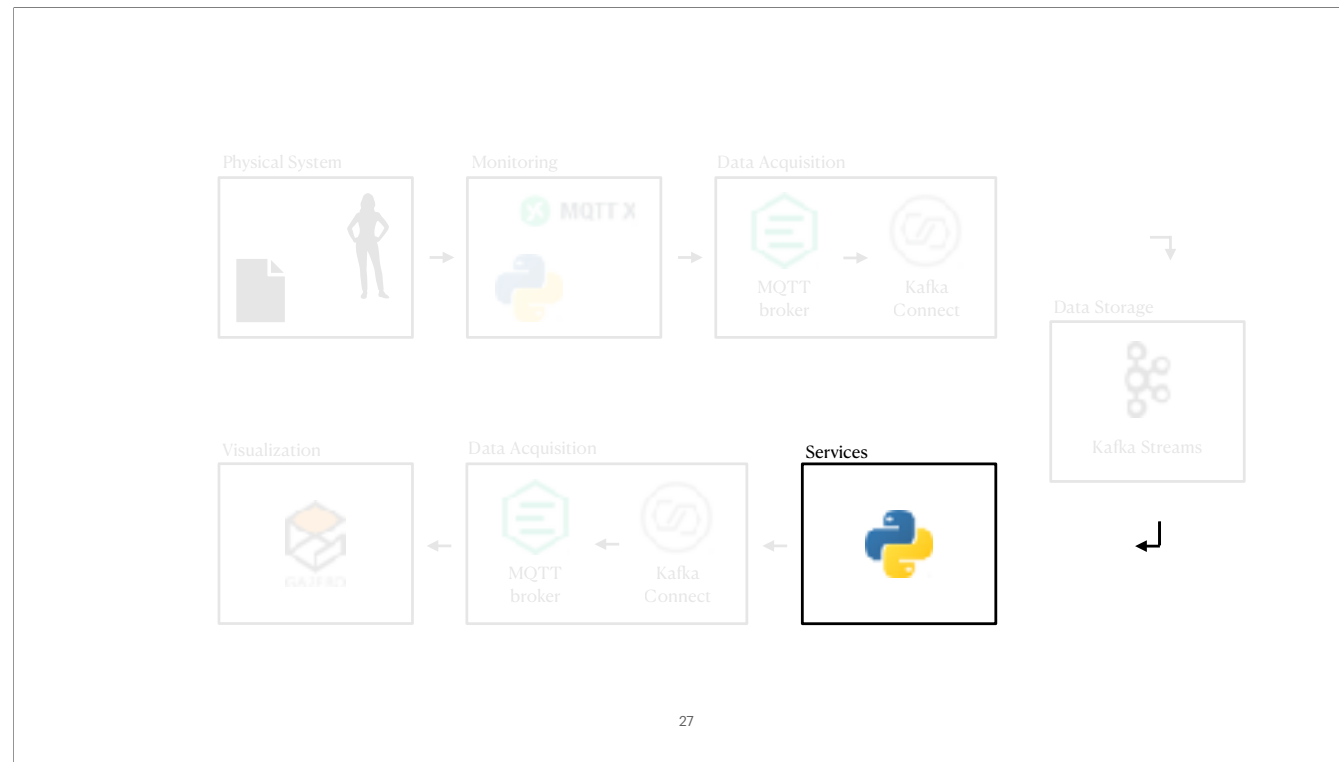
Servico user

E o serviço, que também escuta esse tópico, quando recebe uma mensagem como essa utiliza a ferramenta Ksql para realizar consultas com sintaxe parecida com o SQL no stream de dados dos equipamentos.

Ele obtém então, para cada aparelho, a média de seu consumo de energia no período indicado pelo usuário na mensagem. Seguindo a mesma regra de conversão de potência para cor, ele converte as médias e publica os resultados nos tópicos de saída que chegam ao Gazebo.

Se o usuário deseja voltar à visualização em tempo real ele publica outra mensagem no tópico MQTT com essa intenção. O programa em python em serviços também controla os dois modos de operação, em tempo real e tempo limitado, já que os dados dos sensores nunca param de ser gerados, apenas são ignorados quando executa-se a visualização em tempo limitado.



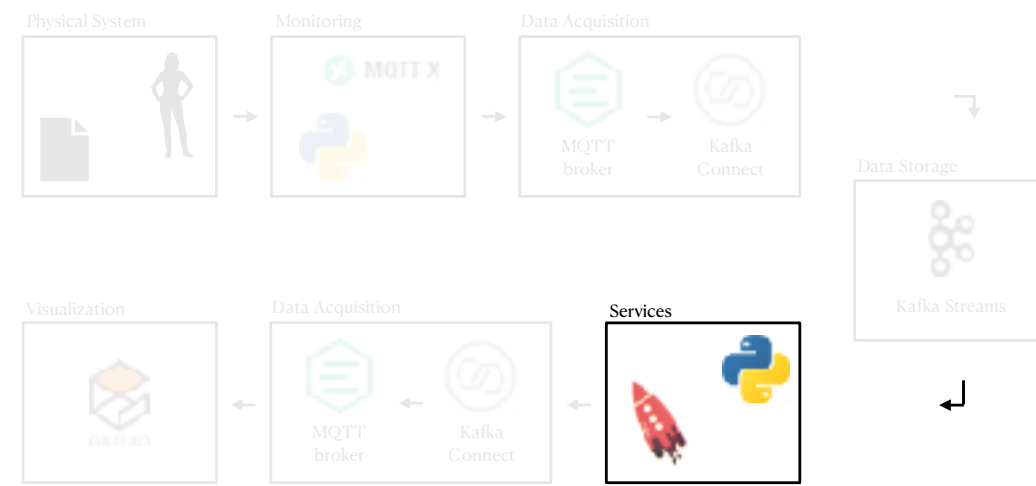


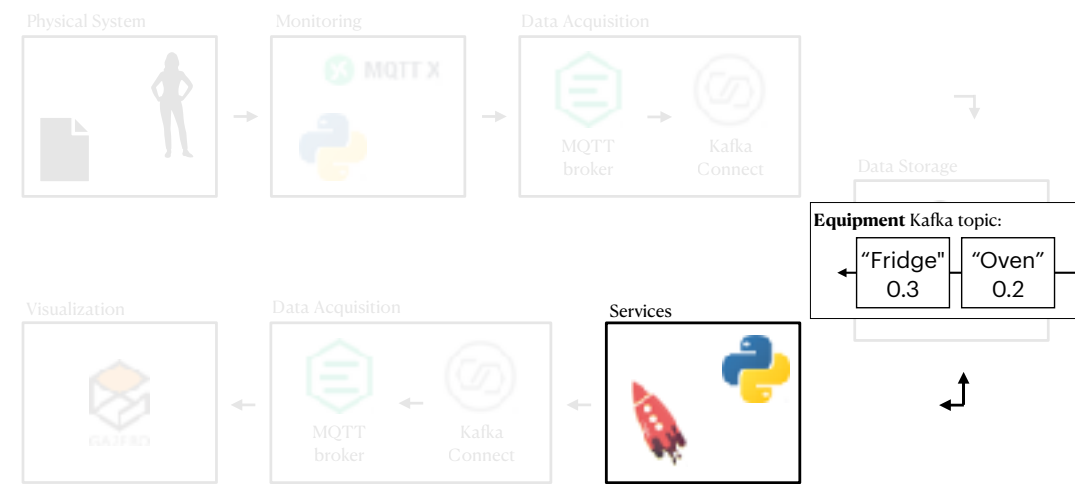
Servico user KSQL

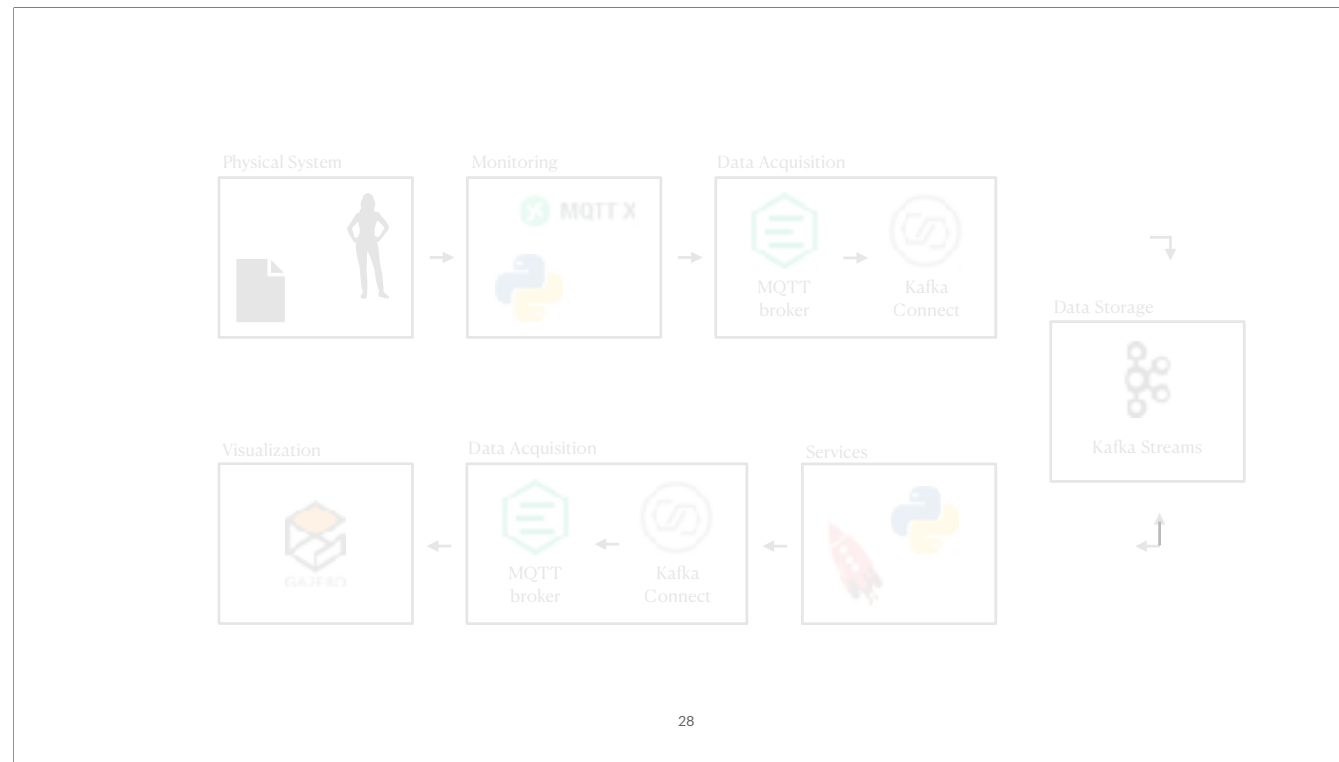
E o serviço, que também escuta esse tópico, quando recebe uma mensagem como essa utiliza a ferramenta Ksql para realizar consultas com sintaxe parecida com o SQL no stream de dados dos equipamentos.

Ele obtém então, para cada aparelho, a média de seu consumo de energia no período indicado pelo usuário na mensagem. Seguindo a mesma regra de conversão de potência para cor, ele converte as médias e publica os resultados nos tópicos de saída que chegam ao Gazebo.

Se o usuário deseja voltar à visualização em tempo real ele publica outra mensagem no tópico MQTT com essa intenção. O programa em Python em serviços também controla os dois modos de operação, em tempo real e tempo limitado, já que os dados dos sensores nunca param de ser gerados, apenas são ignorados quando executa-se a visualização em tempo limitado.







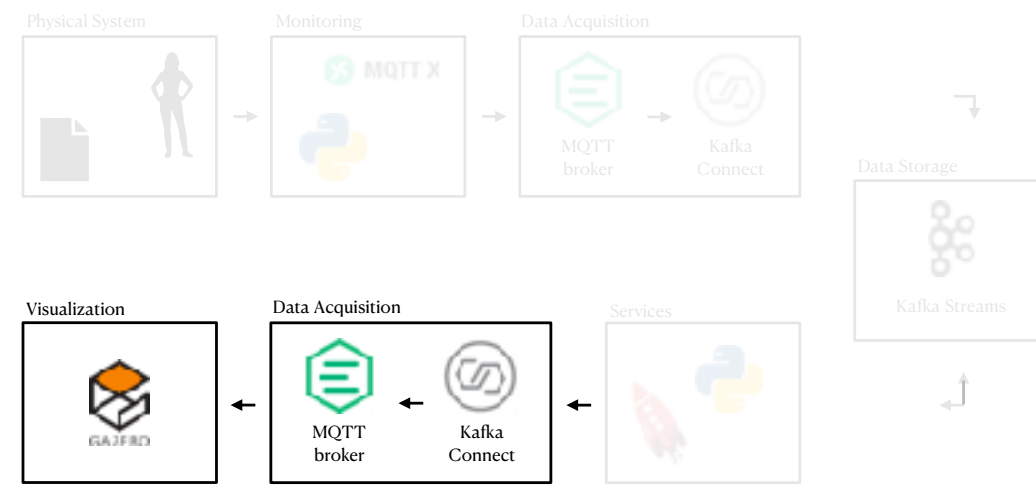
Servico user KSQL

E o serviço, que também escuta esse tópico, quando recebe uma mensagem como essa utiliza a ferramenta Ksql para realizar consultas com sintaxe parecida com o SQL no stream de dados dos equipamentos.

Ele obtém então, para cada aparelho, a média de seu consumo de energia no período indicado pelo usuário na mensagem. Seguindo a mesma regra de conversão de potência para cor, ele converte as médias e publica os resultados nos tópicos de saída que chegam ao Gazebo.

Se o usuário deseja voltar à visualização em tempo real ele publica outra mensagem no tópico MQTT com essa intenção. O programa em Python em serviços também controla os dois modos de operação, em tempo real e tempo limitado, já que os dados dos sensores nunca param de ser gerados, apenas são ignorados quando executa-se a visualização em tempo limitado.

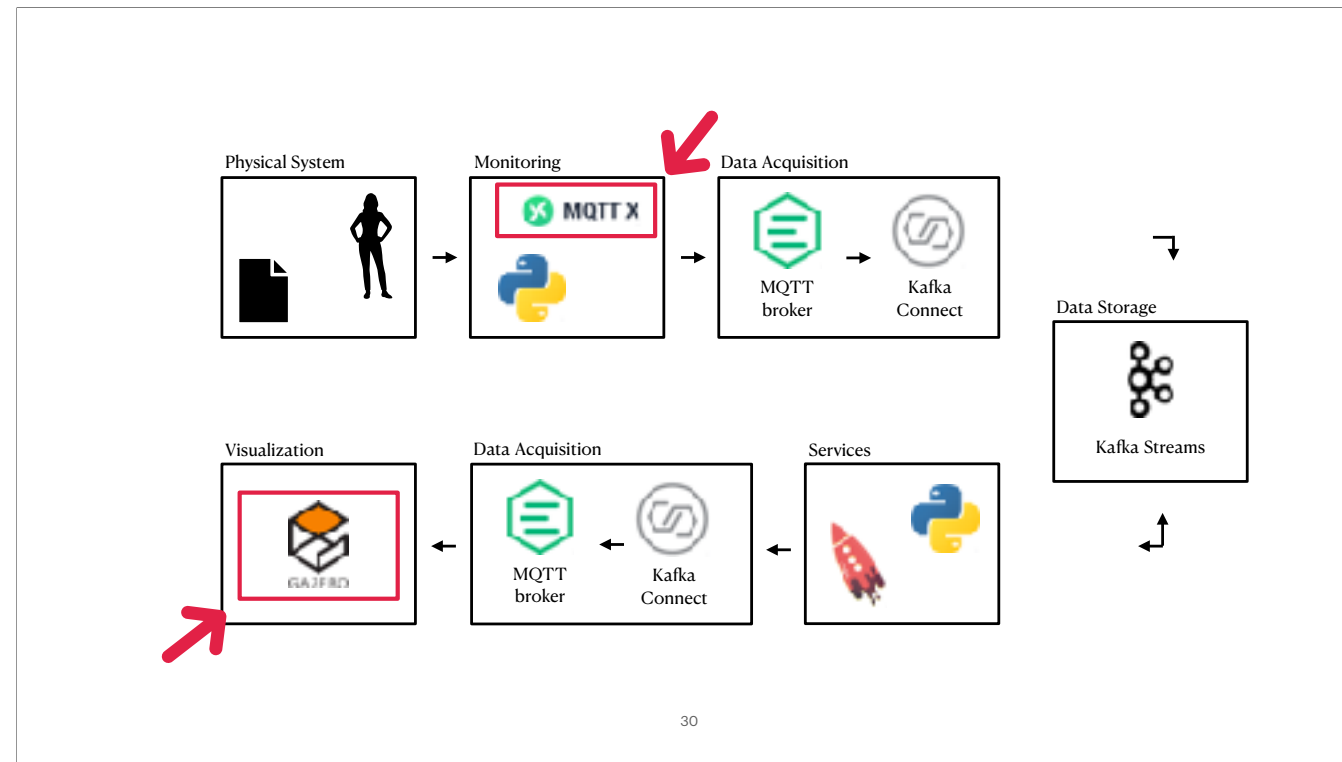
O BROKER É O MESMO



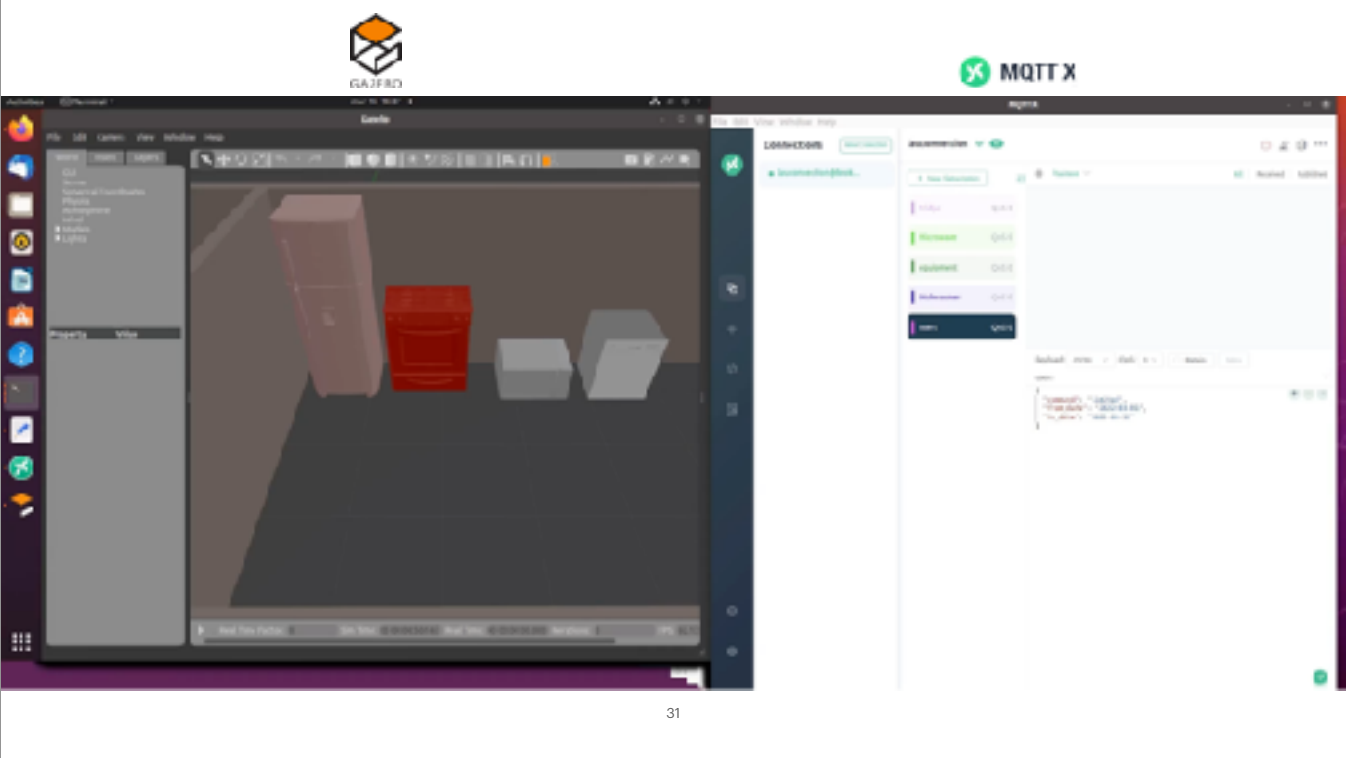
Summary

1. Introduction
2. **Use Case 1**
 1. Introduction
 2. Architecture
 3. **Demo**
3. Use Case 2
 1. Introduction
 2. Architecture
 3. Evaluation
4. Limitations
5. Conclusion

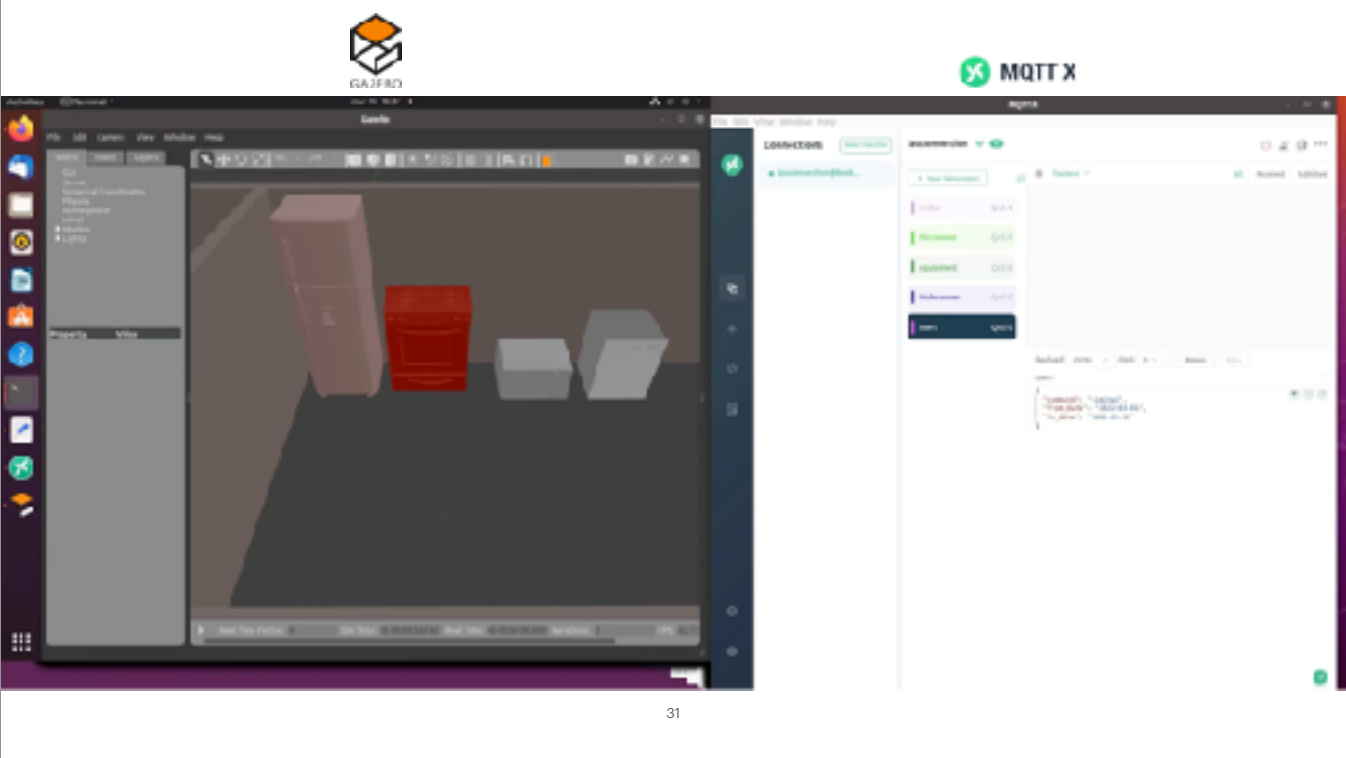
Como implementamos isso



Retomando. Essa é a arquitetura completa e os componentes mostrados são esses dois



demo



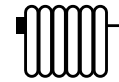
Summary

1. Introduction
2. Use Case 1
 1. Introduction
 2. Architecture
 3. Demo
3. **Use Case 2**
 1. **Introduction**
 2. Architecture
 3. Evaluation
4. Limitations
5. Conclusion

Partindo agora pro caso de uso 2

Use Case 2

Predictive Role

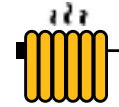


33

O segundo caso de uso foca no papel preditivo do gêmeo digital. Ele busca determinar qual o melhor horário do dia para ligar um aquecedor
clica
de forma que a casa esteja numa temperatura ideal
Clica
na hora que o usuário volta pra casa ao fim do dia.
A gente define que essa temperatura ideal seja 20 graus celsius
clica
e que o usuário fique fora de casa das 8 as 18.

Use Case 2

Predictive Role

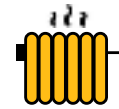


34

O segundo caso de uso foca no papel preditivo do gêmeo digital. Ele busca determinar qual o melhor horário do dia para ligar um aquecedor
clica
de forma que a casa esteja numa temperatura ideal
Clica
na hora que o usuário volta pra casa ao fim do dia.
A gente define que essa temperatura ideal seja 20 graus celsius
clica
e que o usuário fique fora de casa das 8 as 18.

Use Case 2

Predictive Role



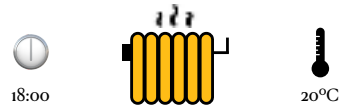
Use Case 2

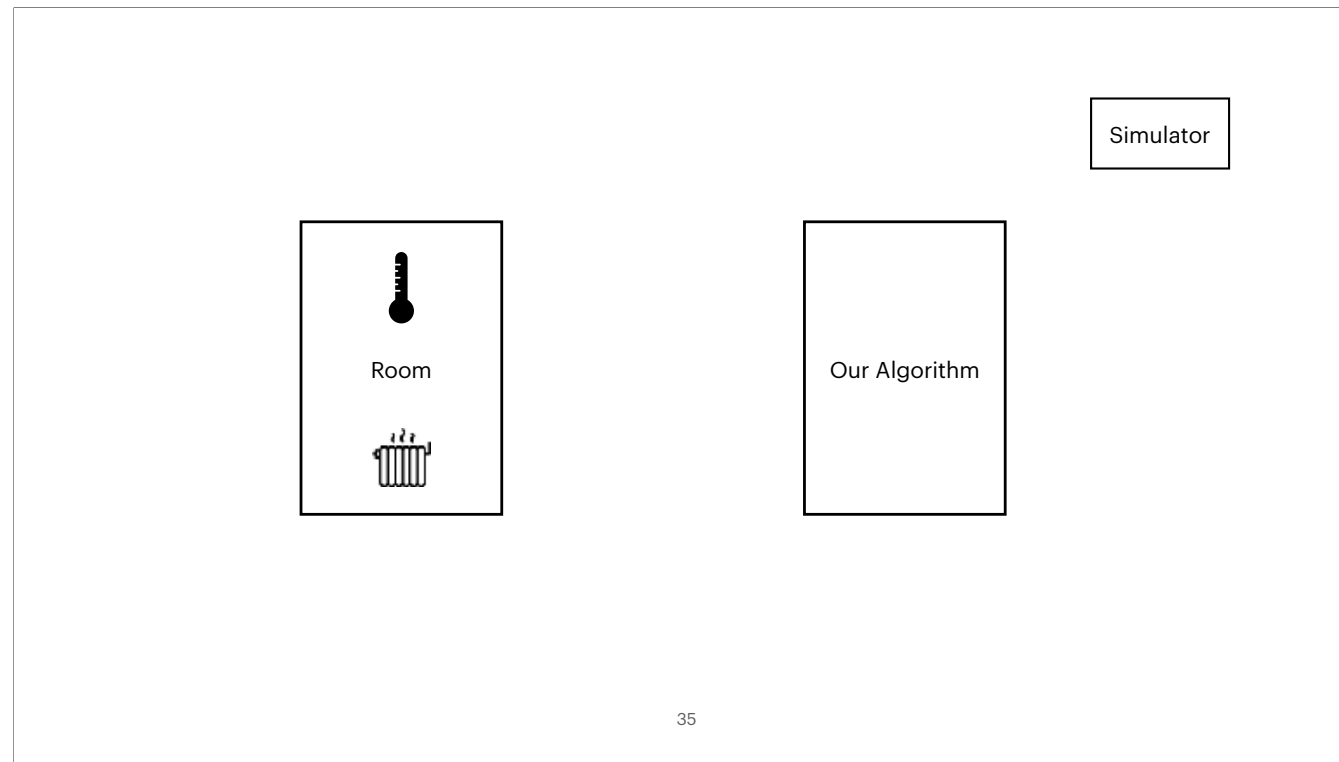
Predictive Role



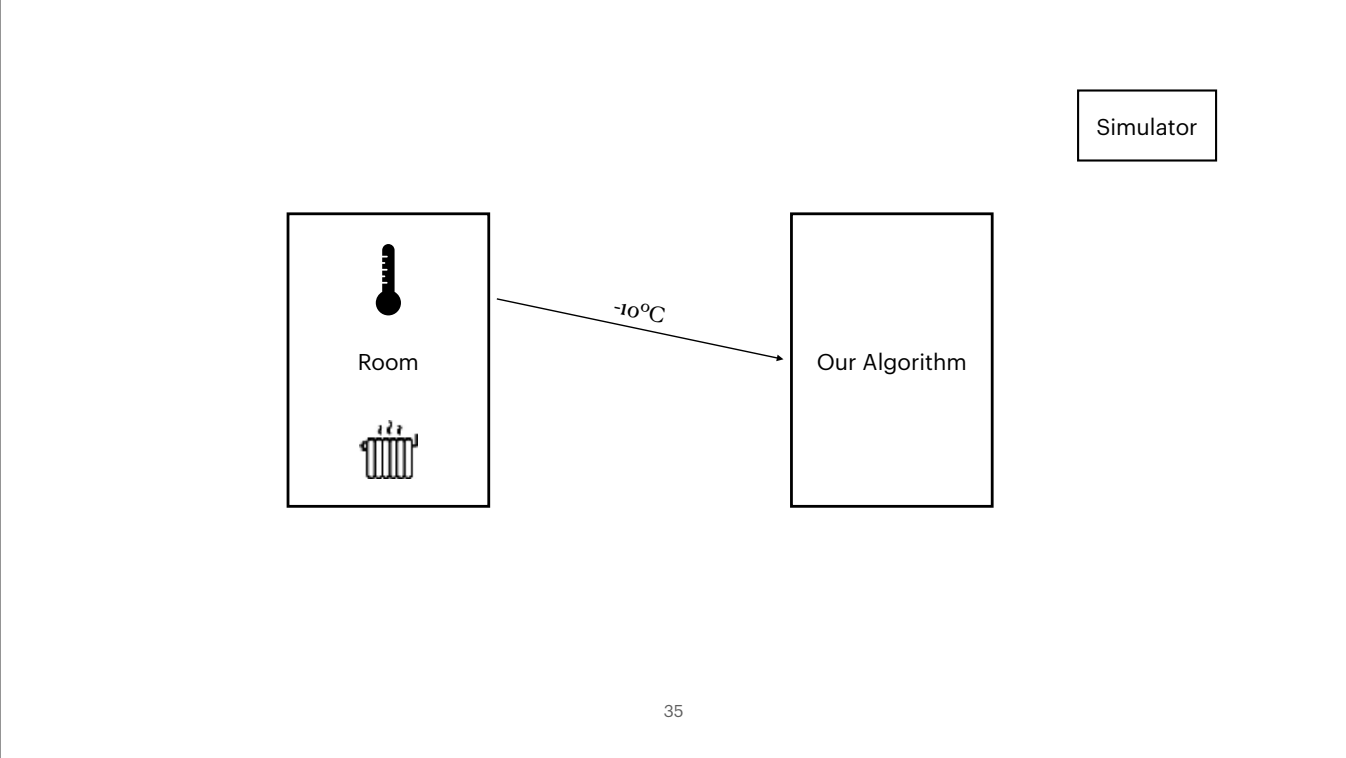
Use Case 2

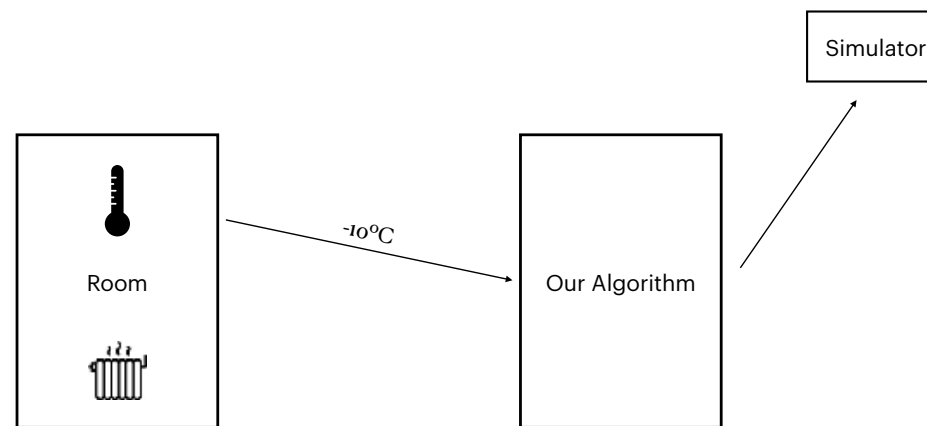
Predictive Role

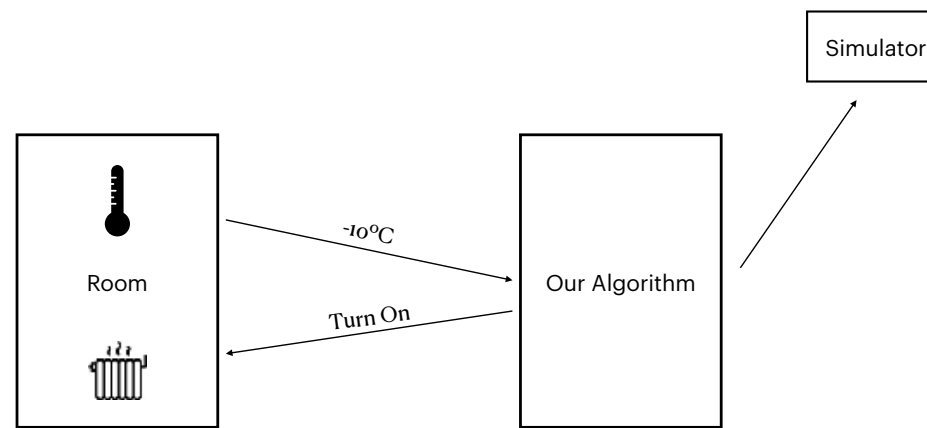




A ideia é que o sistema receba mediadas da temperatura interna de uma casa. Simule o aquecimento da casa do horário em que a medida foi feita até as 18h e determine se é necessário ou não ligar o aquecedor.









Essa simulação foi feita com a ajuda de um software, chamado de Energy 2D, que nos permite simular o aquecimento de um cômodos. Podemos colocar componentes como, a janela, mais à esquerda, fonte de calor, em amarelo, e termômetros, incluindo termostatos, que controlam o aquecedor sendo conectados a ele. O outro, localizado mais longe do ambiente, é o determina a temperatura do ambiente.



37

Ao longo da simulação, os quartos vão aquecendo.

Time	T1	T2
100.0	0.644777618	0.604433838
200.0	0.29112577	0.675362734
300.0	0.5854305	0.2128045
400.0	0.84571147	0.16267102
500.0	1.0000641	0.50016475
600.0	1.2328494	0.61893183
700.0	1.3712568	0.71930534
800.0	1.482081	0.4035421
900.0	1.5710008	0.47417655
1000.0	1.6425635	0.93350863
1100.0	1.7003437	0.9834862
1200.0	1.7471402	1.0257179
1300.0	1.7851415	1.0615181
1400.0	1.8160657	1.0919617
1500.0	1.8412676	1.1179292
1600.0	1.861823	1.1401435
1700.0	1.8785893	1.1592016
1800.0	1.8922533	1.1755996
1900.0	1.9033700	1.1897497
2000.0	1.9123893	1.2019978
2100.0	1.9196769	1.2126341
2200.0	1.9255314	1.2219032
2300.0	1.9301993	1.2300116
2400.0	1.9338821	1.2371337
2500.0	1.9367486	1.2434186
2600.0	1.9389339	1.2489917
2700.0	1.9405596	1.2539614
2800.0	1.9417169	1.2584182
2900.0	1.9424865	1.2624404
3000.0	1.9429356	1.2660946

38

Ao final, a gente tem uma saída como esta. Que mostra as medidas de todos os termômetros ao longo do tempo.

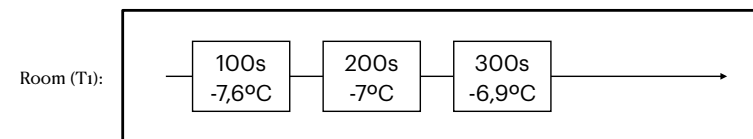
Consideramos então a medida do termómetro mais afastado como o que realmente representa a temperatura do ambiente.

E passamos essa progressão temporal para um stream no Kafka.

Time	T1	T2
100.0	0.644777618	0.604433838
200.0	0.29112577	0.675362734
300.0	0.5854305	0.2128045
400.0	0.84571147	0.16267102
500.0	1.0600641	0.50016475
600.0	1.2328494	0.61893183
700.0	1.3712568	0.71930534
800.0	1.482081	0.4035421
900.0	1.5710008	0.47417655
1000.0	1.6425635	0.93350863
1100.0	1.7003437	0.9834862
1200.0	1.7471402	1.0257179
1300.0	1.7851415	1.0615181
1400.0	1.8160657	1.0919617
1500.0	1.8412676	1.1179292
1600.0	1.861823	1.1401435
1700.0	1.8785893	1.1592016
1800.0	1.8922533	1.1755996
1900.0	1.9033700	1.1897497
2000.0	1.9123893	1.2019978
2100.0	1.9196769	1.2126341
2200.0	1.9255314	1.2219032
2300.0	1.9301993	1.2300116
2400.0	1.9338821	1.2371337
2500.0	1.9367486	1.2434186
2600.0	1.9389339	1.2489917
2700.0	1.9405596	1.2539614
2800.0	1.9417169	1.2584182
2900.0	1.9424865	1.2624404
3000.0	1.9429356	1.2660946







Summary

1. Introduction
2. Use Case 1
 1. Introduction
 2. Architecture
 3. Demo
3. **Use Case 2**
 1. Introduction
 2. **Architecture**
 3. Evaluation
4. Limitations
5. Conclusion

Partindo agora pra implementação

A implementacao segue o mesmo padrao do primeiro caso de uso.

Idealmente teriamos uma casa com termometros, aquecedores e tomadas smart para que pudesseamos ligar os aquecedores de acordo com o nosso algoritmo.

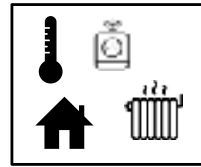
De novo, simulamos isso com um conjunto de dados já pronto.

Ele contém uma medicao de temperatura por hora durante um ano de uma cidade alemã que eu fiz intercambio, Stuttgart.

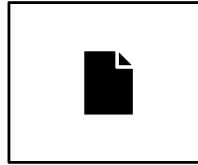
Physical System



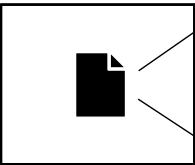
Physical System



Physical System



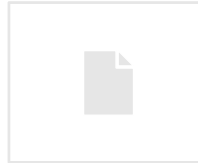
Physical System



Hour	Date	Temperature
1	1/1	-5
2	1/1	-4
...	...	-3
24	1/1	-1
1	2/1	3

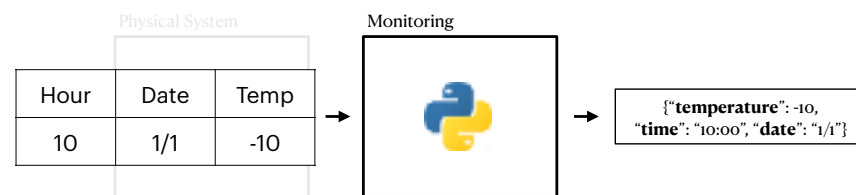


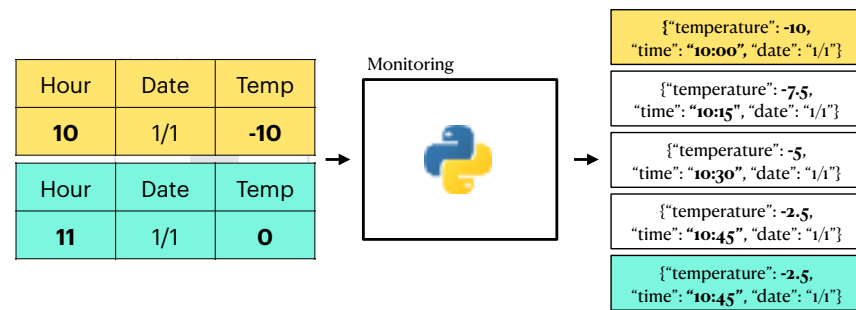
Physical System



Temos então um script que lê esse arquivo CSV linha por linha, transforma cada uma em uma mensagem JSON.

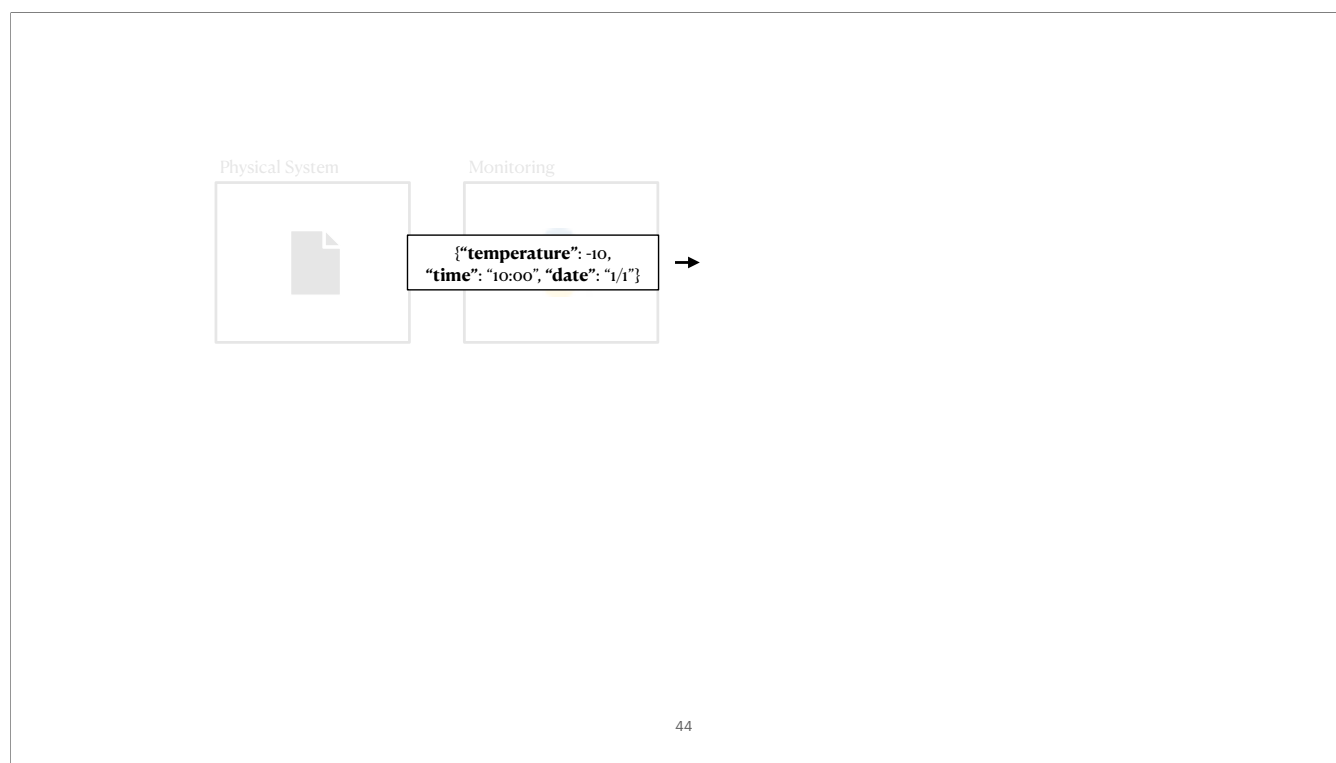
Pra aumentar a precisão do sistema, esse script gera 3 valores entre cada duas linha, simulando um sistema fisico que gera valores de temperatura a cada 15 minutos ao inves de uma hora.





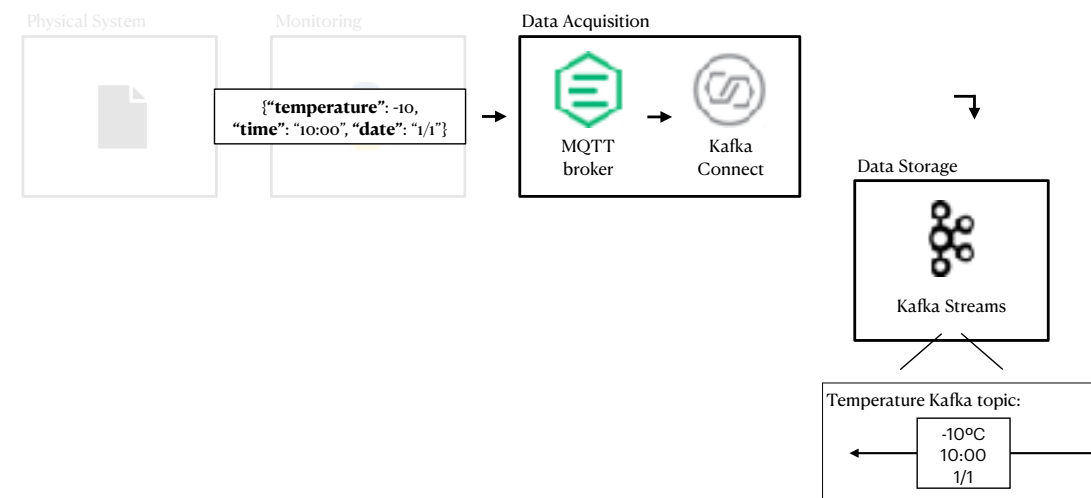
Temos então um script que lê esse arquivo CSV linha por linha, transforma cada uma em uma mensagem JSON.

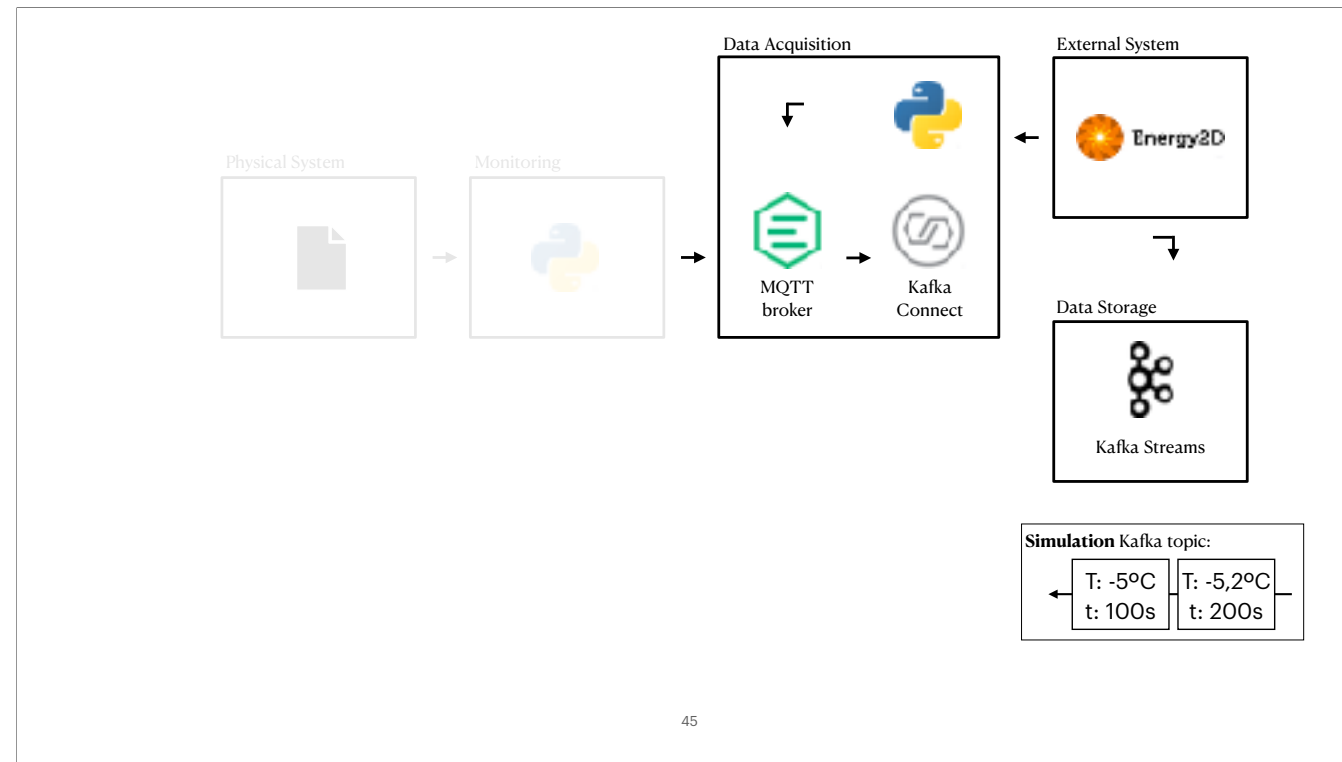
Pra aumentar a precisão do sistema, esse script gera 3 valores entre cada duas linha, simulando medidas de 15 em 15 minutos ao inves de uma hora. Esses valores seguem uma progressao linear.



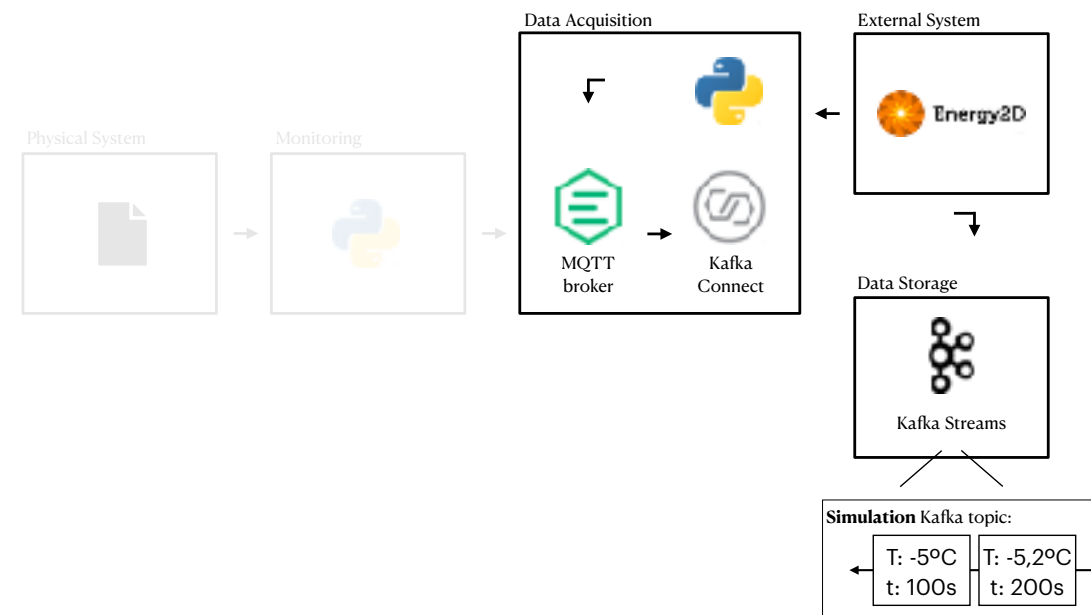
O script entao publica cada mensagem em um topico MQTT

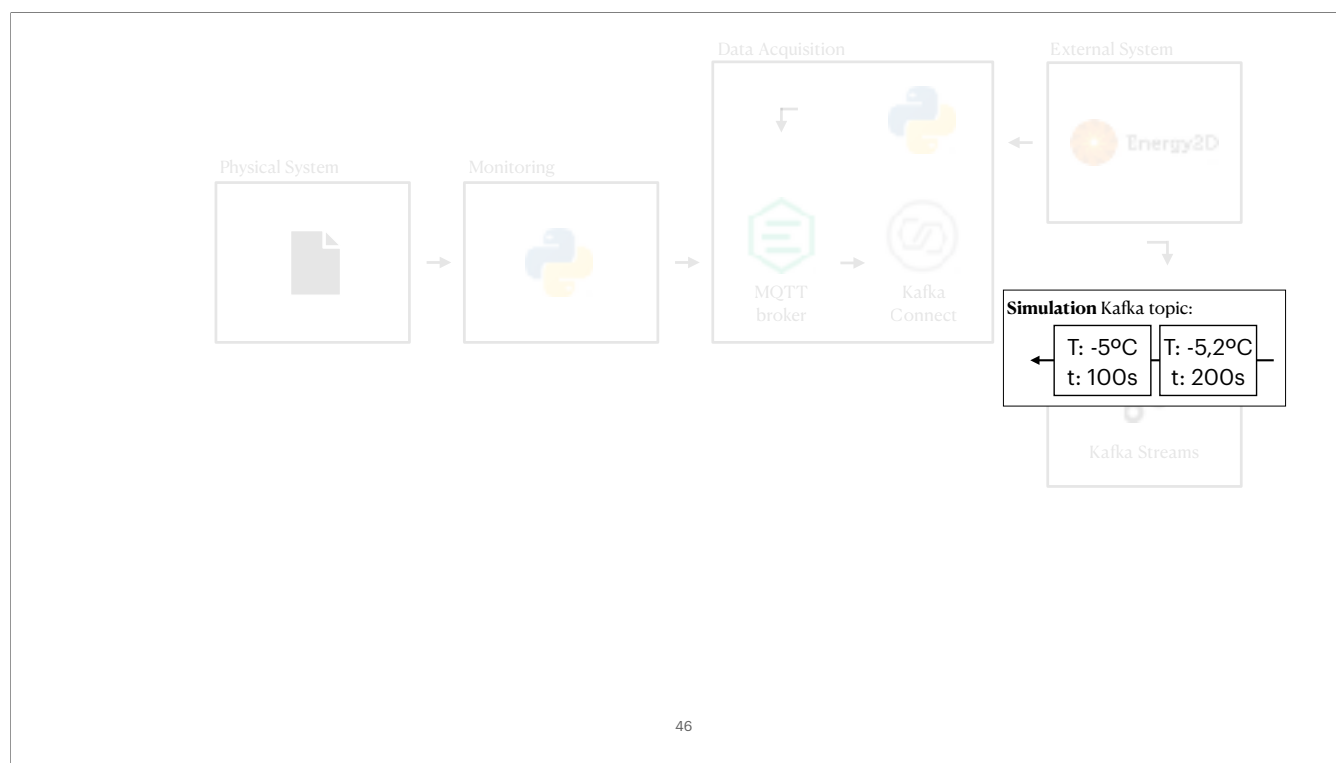
E elas novamente passam pelo Connect e sao armazenadas em um stream no Kafka.



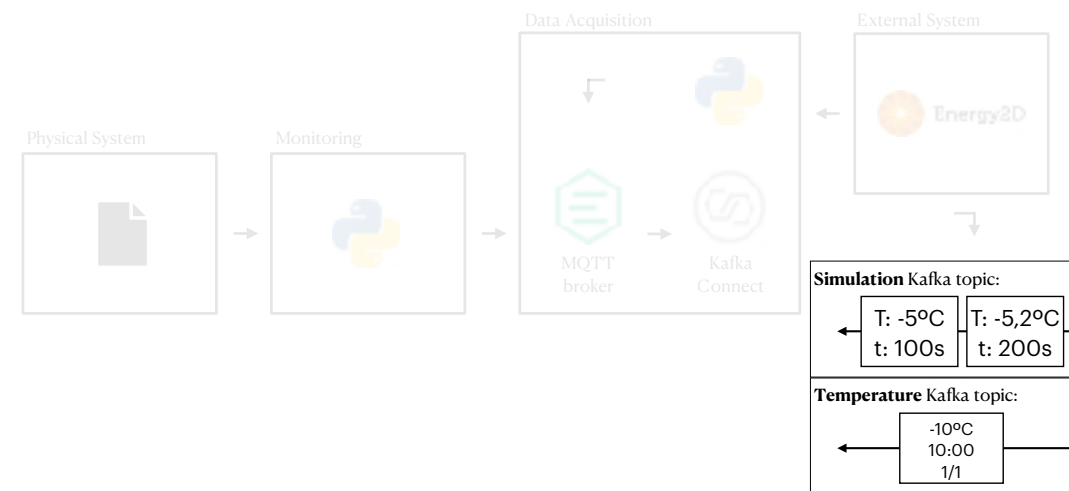


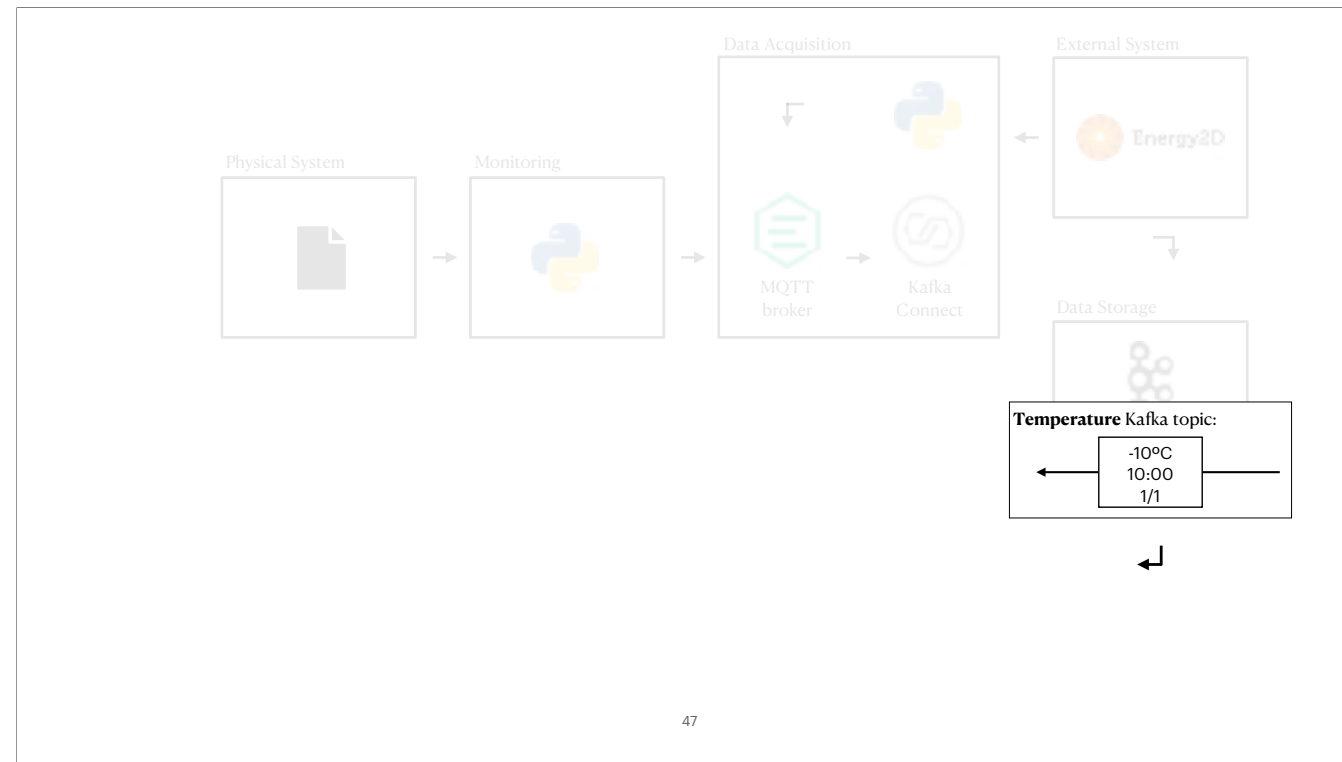
Esse use case conta com um script adicional que lê os arquivos CSV gerados pelo simulador e publica cada mensagem em um outro tópico MQTT, que passam pelo connect e terminam em um novo stream no Kafka.





Temos então 2 tópicos no Kafka. Um que recebe dados do ambiente físico a cada 15 minutos e outro os dados da simulação.





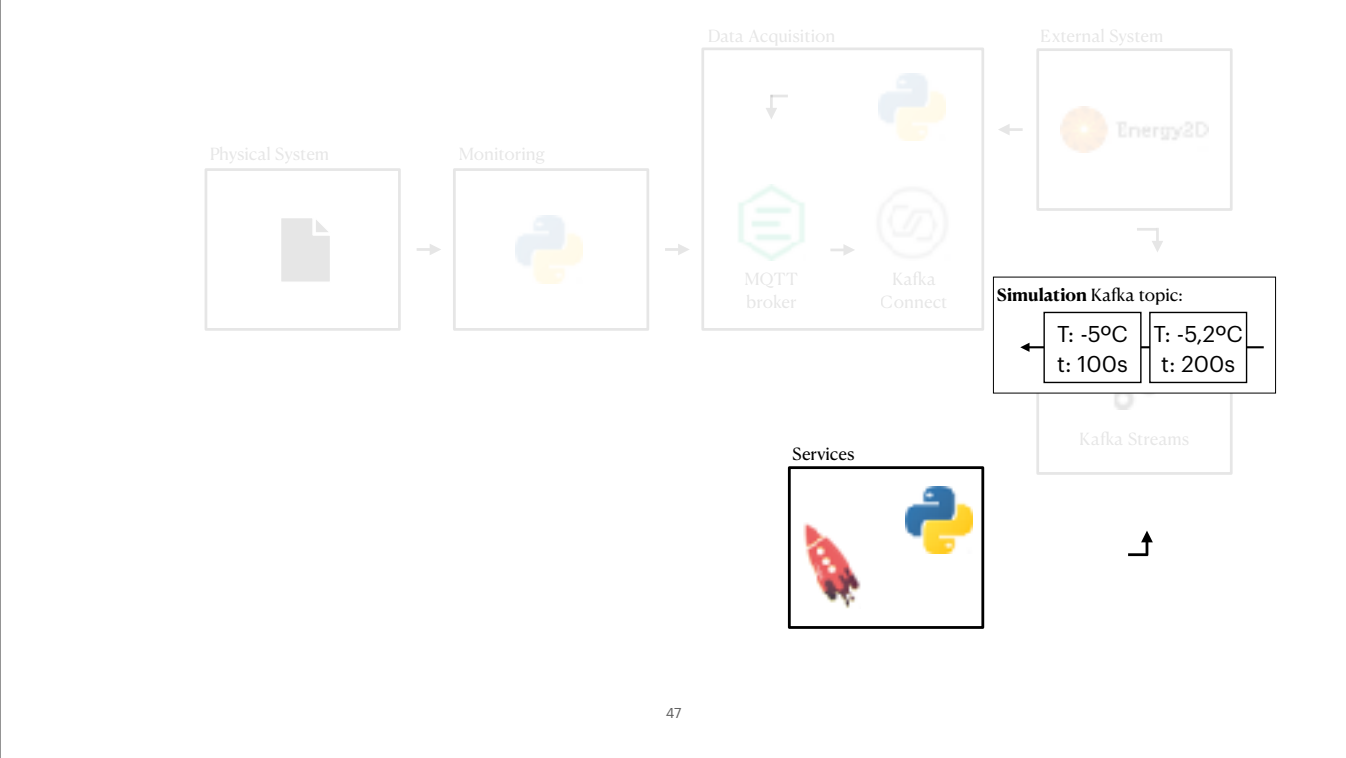
Um terceiro script o topico com mensagens provenientes do sistema fisico e

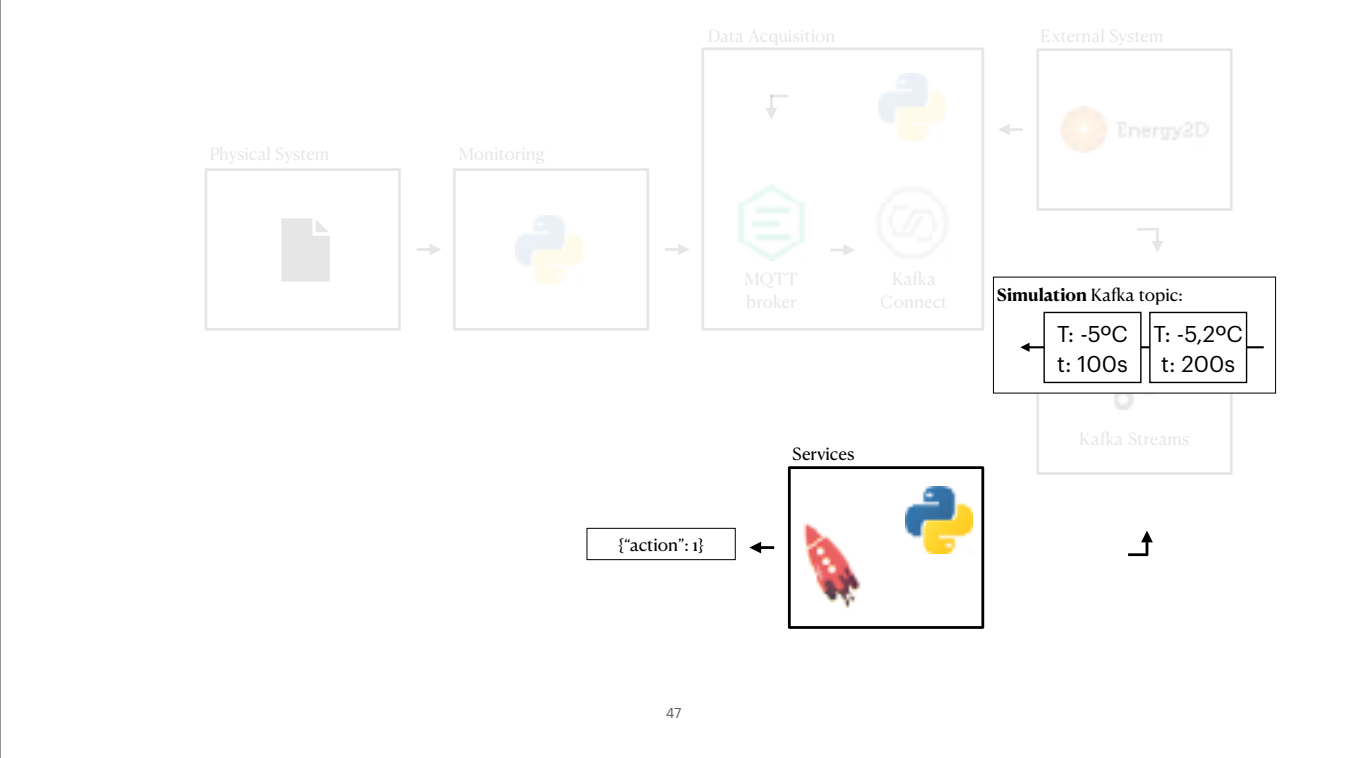
realiza consultas kSQL ao stream da simulação

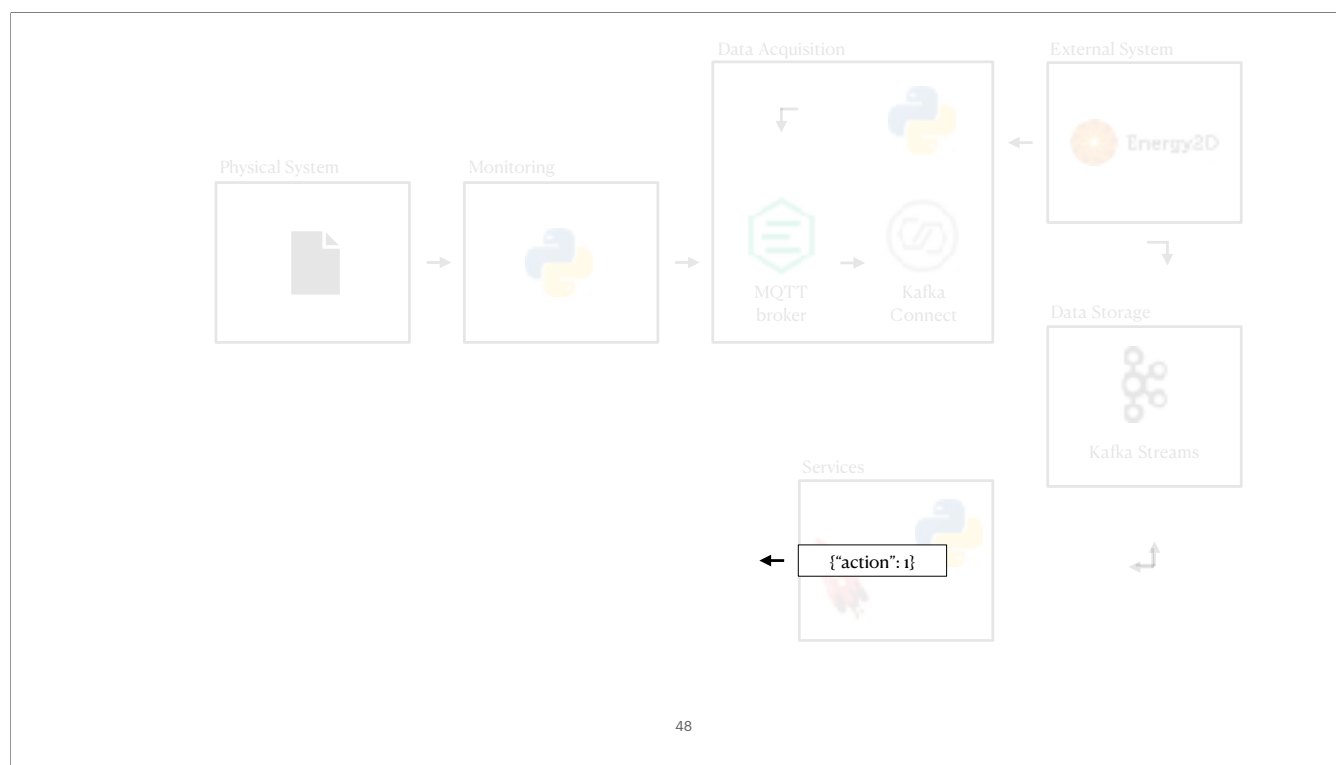
Assim, determina se o aquecedor deve ser ligado ou nao.

Representa essa decisão como uma mensagem JSON contendo um booleano

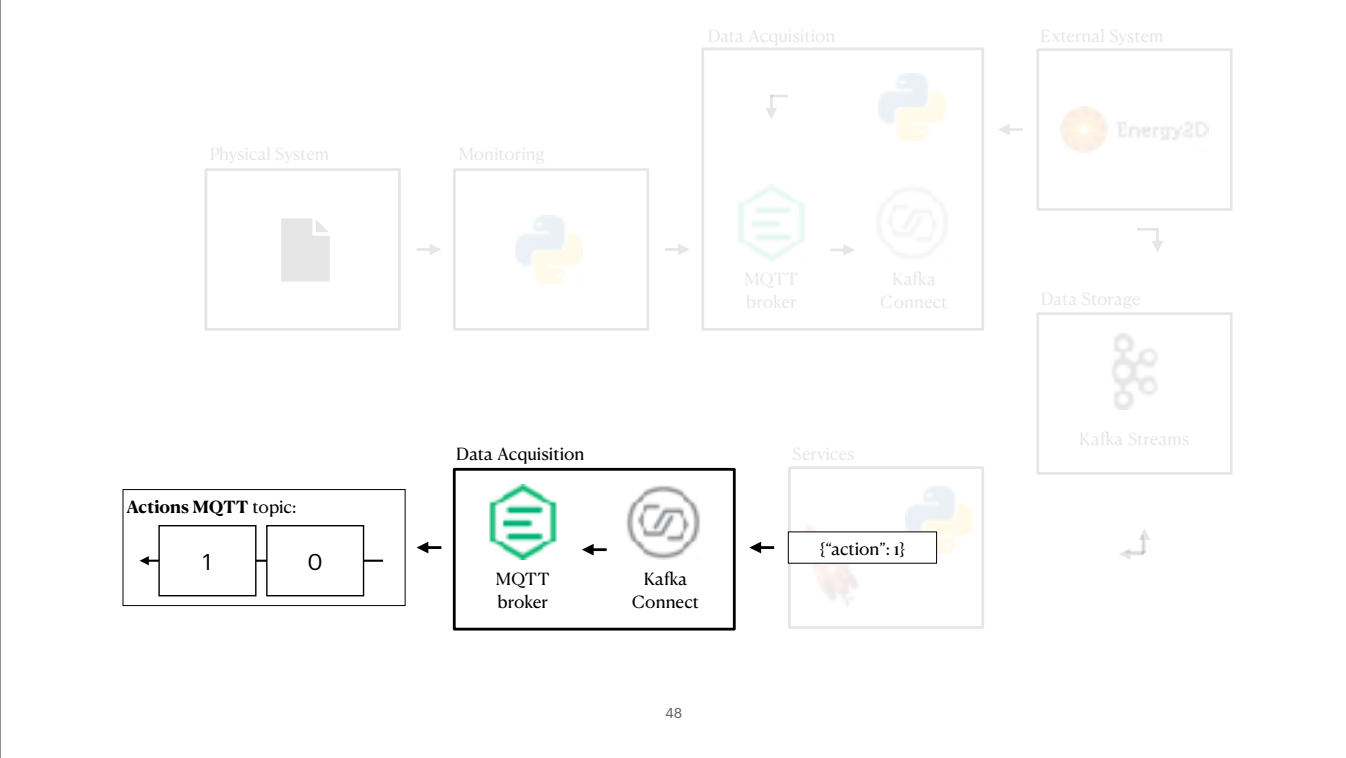


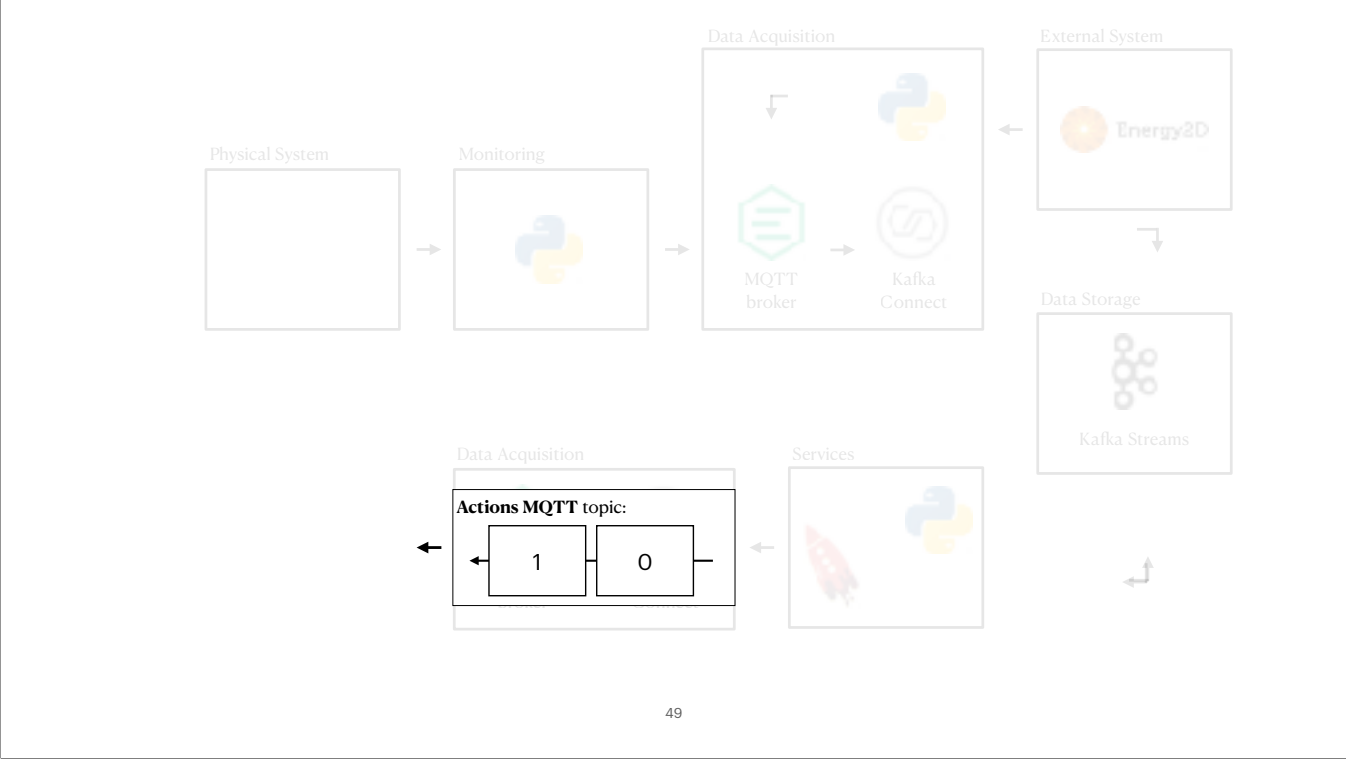






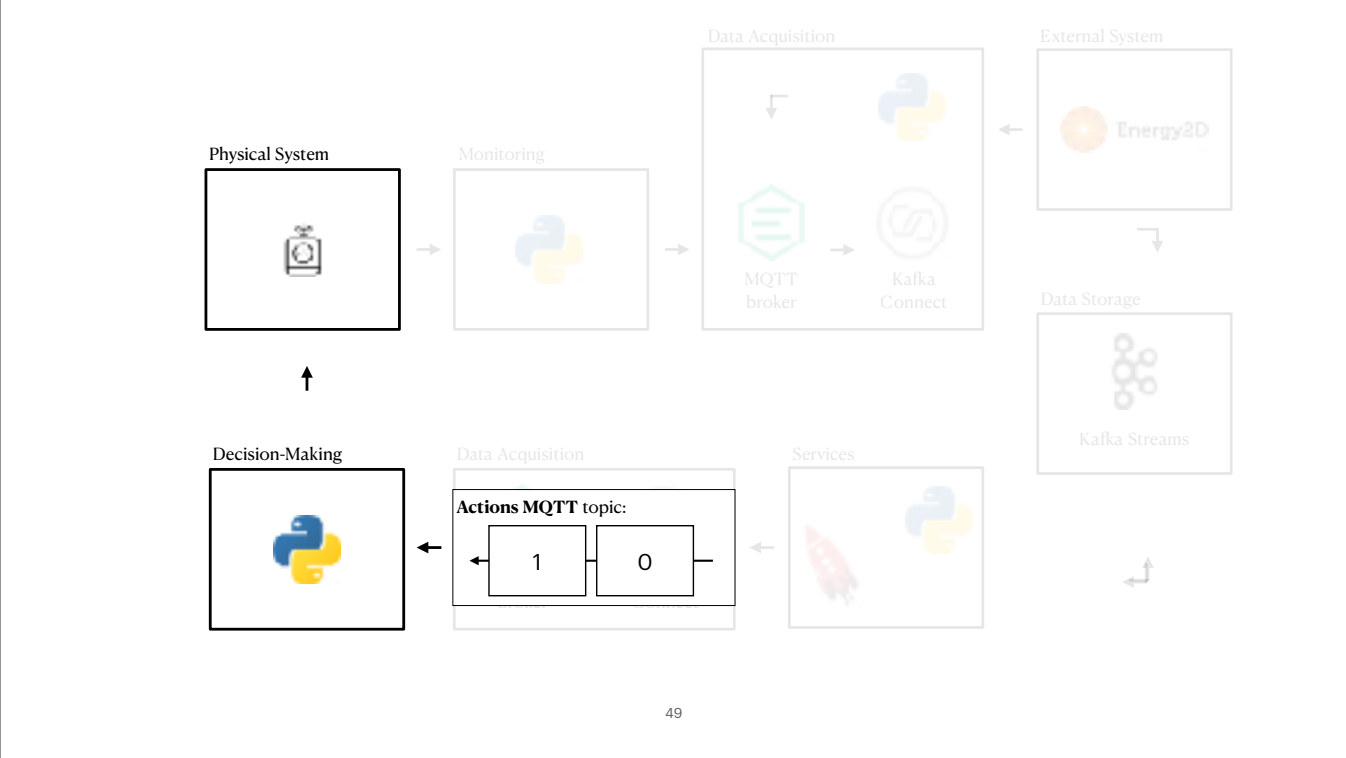
Publica a mensagem entao a em um novo topico Kafka,
que é repassada a um topico MQTT

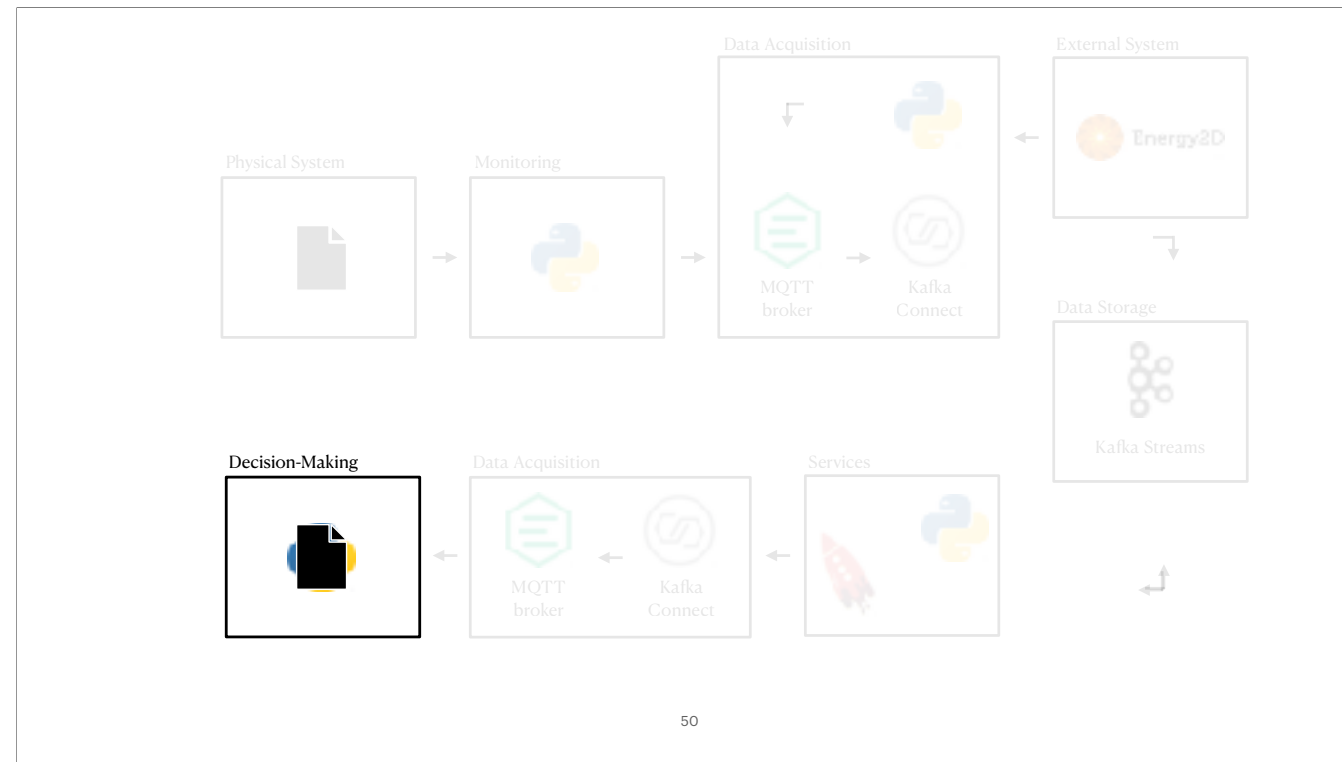




DECISION MAKING

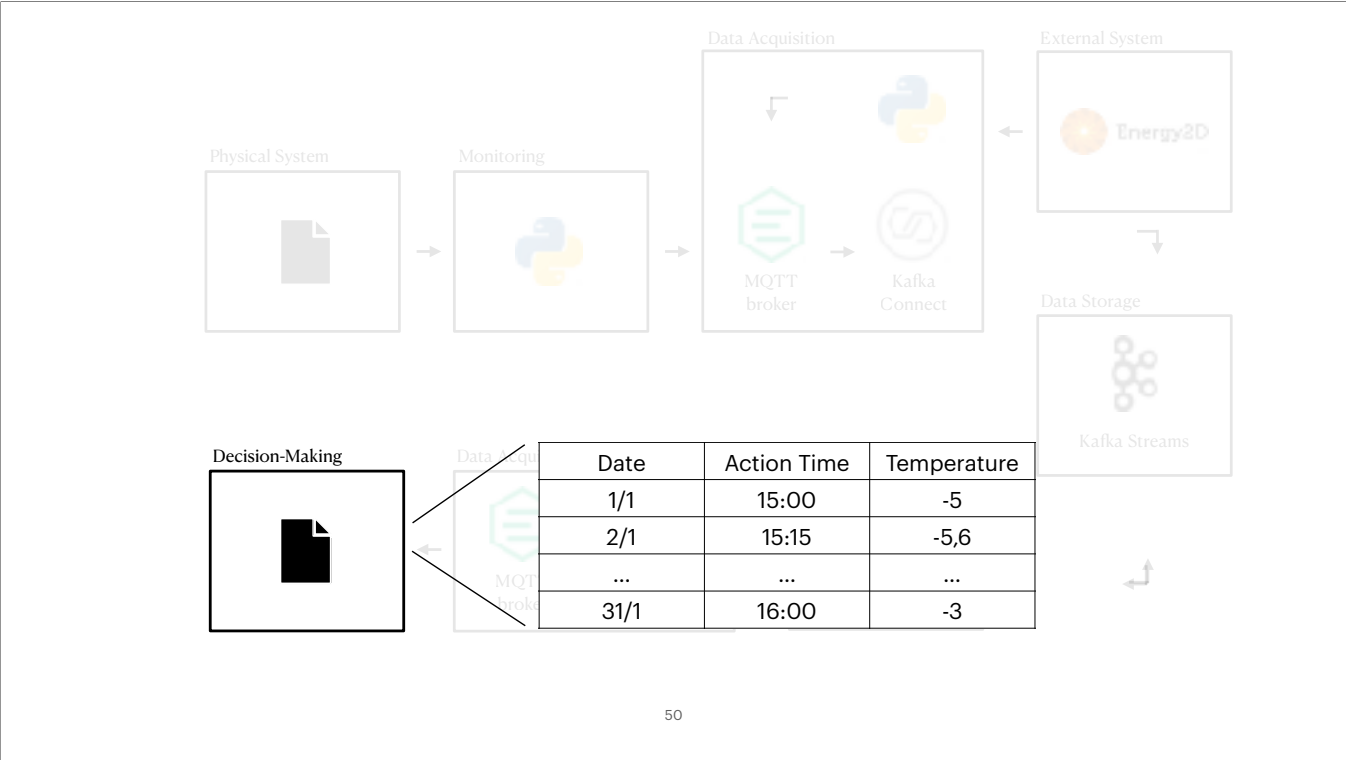
Idealmente teriamos um consumidor dessas acoes que iria passar o comando pra tomada.





DECISION MAKING

Como não temos um sistema físico, esse script, na prática ele é o mesmo que o de monitoramento, tem como saída um arquivo CSV que associa aos dias em que a iteração foi executada, a hora o aquecedor foi ligado e em que temperatura o cômodo estava naquele momento, para depois avaliarmos o sistema.



Summary

1. Introduction
2. Use Case 1
 1. Introduction
 2. Architecture
 3. Demo
3. **Use Case 2**
 1. Introduction
 2. Architecture
 3. **Evaluation**
4. Limitations
5. Conclusion

Evaluation

4 months

3 rooms

Jan

small-room

Fev

big-room

Mar

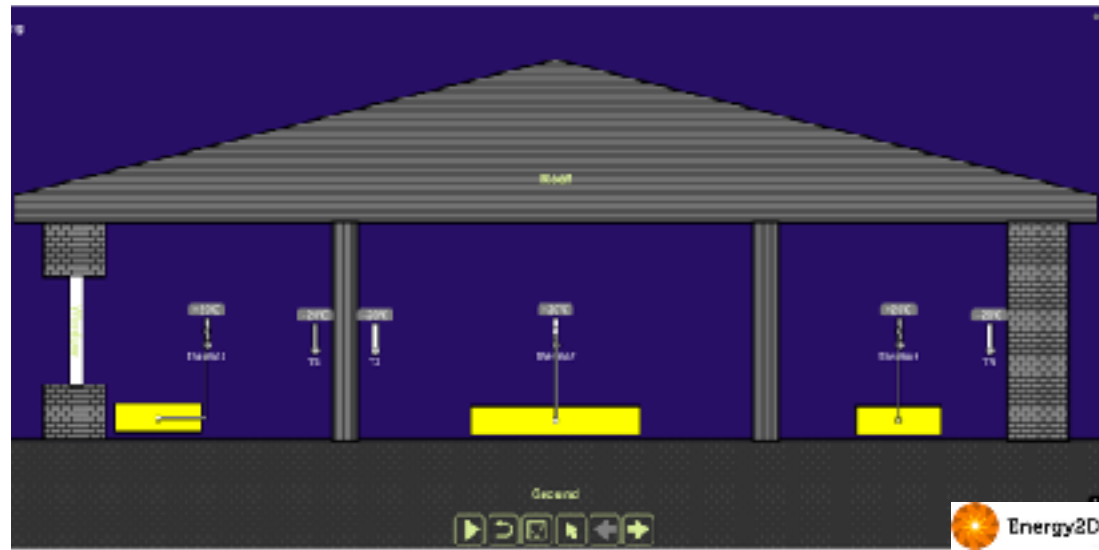
small-room-window

Apr

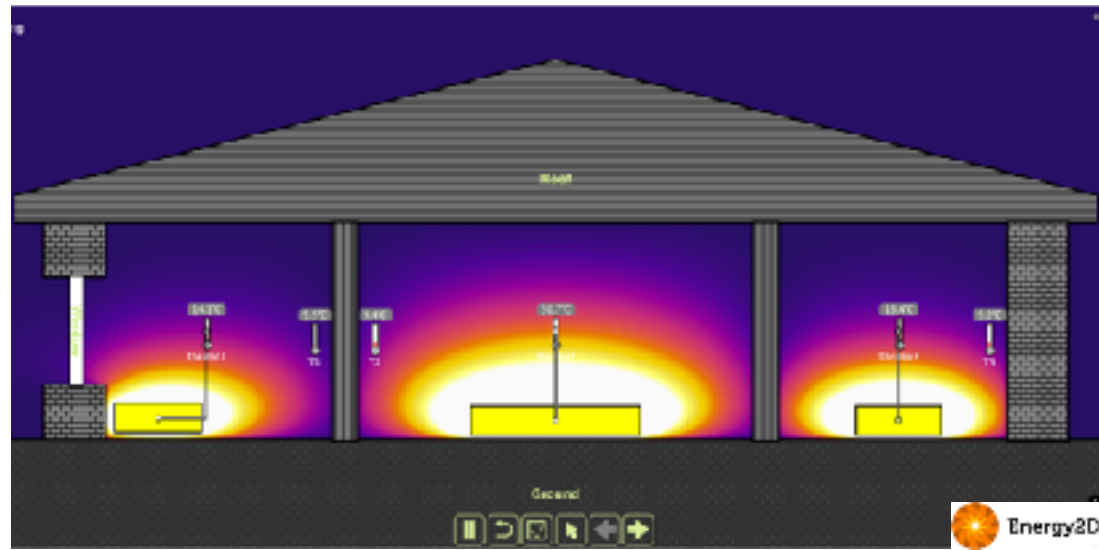
x

Falando então sobre a avaliação.

Medimos o funcionamento do sistema nos 4 primeiros meses do ano e em 3 quartos diferentes. Um pequeno, um maior e outro do mesmo tamanho que o pequeno mas com janela.



Então no Energy 2D criamos esses 3 quartos. De novo, o termómetro localizado mais longe do ambiente, é o determina a temperatura do ambiente.



54

Rodamos por 13 horas.

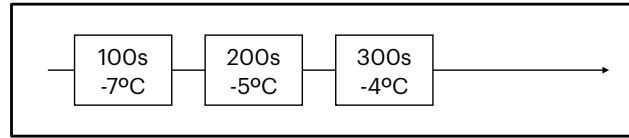
Time	Therm1	T2	Therm3	Therm2	T5	T1
1000.0	104630992	7.02331138	16.0333958	15.0423427	6.06006085	6.0000035
2000.0	17.1141394	6.052294	11.0862391	10.0626231	4.61071883	-6.64813904
3000.0	11.2486108	6.052164	8.05823	7.12948	1.34209962	2.2622
4000.0	11.472809	3.3380764	6.6739883	4.0353544	2.37623715	2.433336
5000.0	-0.04247	2.7482102	4.064024	1.4666576	1.80134827	1.80134748
6000.0	-6.943908	2.3421108	3.03660527	2.3834364	1.39623762	1.3448233
7000.0	-1.110802	1.7044218	2.04339902	1.6370544	1.08088127	-0.0606862
8000.0	-3.478222	-1.388424	-1.3249831	-0.8088090	-0.8798940	-0.62585896
9000.0	-1.7949826	-1.0250602	-0.70387896	-0.171576022	-0.622005105	-0.3343560
10000.0	-0.62184876	-0.6021291	-0.32343604	0.4752051	-0.42210550	-0.04004212
11000.0	3.6699231	-0.32405105	0.40200243	1.010752	-0.23191322	0.24017552
12000.0	1.5814782	0.6040238127	0.51510435	1.5005002	-0.046053792	0.5305010
13000.0	3.6882822	0.32379896	1.4241545	2.3400020	0.13044134	0.0107002
14000.0	8.24132	0.6420361	1.5215400	2.905334	0.33000190	1.002002
15000.0	5.2689675	0.5000169	2.4125002	3.2573300	0.50100102	1.2926471
16000.0	5.4778617	1.2252838	2.5030254	3.0003025	0.60575245	1.6840040
17000.0	7.269926	1.3510884	3.2000105	4.332044	0.8700002	1.970000
18000.0	5.0588415	1.5002582	5.2343302	4.857758	1.0374001	2.271274
19000.0	9.321871	2.2282494	4.523400	5.4433323	1.2433275	2.5800437
20000.0	12.784588	2.540245	4.85522	5.8033342	1.4332034	2.880780
21000.0	11.848783	2.873024	5.300043	6.323203	1.6207203	3.171122
22000.0	11.8989843	3.100124	5.557804	7.000003	1.8300002	3.4700004
23000.0	13.538888	3.3208	6.282504	7.0024032	2.0340005	3.7801
24000.0	14.578831	3.8504198	6.303822	8.14021	2.2334400	4.0304102
25000.0	16.8118071	4.154011	7.44481	8.040048	2.4050558	4.4870806
26000.0	17.8056	4.54111	7.4414288	8.4401905	2.6100019	4.7210055
27000.0	18.8078862	4.841486	8.1892	8.450122	2.810004	5.1100153
28000.0	18.188017	5.1405852	8.881641	10.285855	3.015008	5.4500085
29000.0	20.127876	5.500124	8.180188	10.800124	3.2212085	5.88019011
30000.0	21.147824	5.5000027	8.810181	11.350435	3.42855	6.1811478
31000.0	22.159854	5.202045	10.080101	11.885852	3.6300003	6.300455
32000.0	23.171592	5.812195	10.551128	12.420041	3.8405250	6.68001150
33000.0	24.185156	6.513394	11.030126	12.558055	4.0012473	6.98001650
34000.0	25.184238	7.338488	11.503725	13.403055	4.235702	7.330118
35000.0	26.288843	7.7005886	11.574853	14.035783	4.403030	7.6715073
36000.0	27.28536	8.00307	12.448855	14.550503	4.7000043	8.034027
37000.0	28.287558	8.437727	12.518150	15.00344	4.8303400	8.358005
38000.0	29.287636	8.830742	13.38038	15.827355	5.351300	8.704110

Ao final, a gente tem uma saída como essa. Que mostra as medidas de todos os termômetros ao longo do tempo.

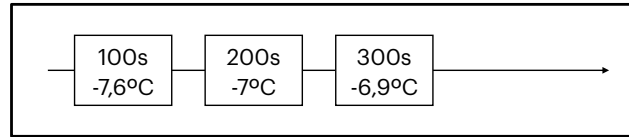
O script de aquisicao de dados passa os dados dos termômetros de interesse para streams no Kafka.

Time	Therm2	T2	Therm2	Therm2	T2	T2
1000.0	10418990	7.0231135	16.033395	15.042347	6.0600685	6.000035
2000.0	17.114394	6.052294	11.086239	10.062623	4.6107183	4.6411364
3000.0	11.288105	4.052164	8.05823	7.12948	3.3429962	3.2827
4000.0	11.472899	3.13389764	4.673983	4.0353544	2.37623715	2.433336
5000.0	0.04247	2.7487107	4.064074	3.4966576	1.80113457	1.8034748
6000.0	-6.943905	2.3421105	3.0366057	3.3834364	1.39623757	1.3448233
7000.0	-1.110897	-1.7044218	2.0433992	1.6270544	1.0808817	0.9466862
8000.0	-3.478722	-1.389424	-1.3249831	-0.8088096	-0.879848	-0.6758586
9000.0	-1.7949826	-1.0250602	-0.70387896	-0.37157622	-0.62200516	-0.3343566
10000.0	-0.62184876	-0.6221291	-0.3343604	0.475251	-0.42210558	-0.04664213
11000.0	3.6699231	-0.32405105	0.40200243	1.030757	-0.23191322	0.24017551
12000.0	1.5814787	0.6040238127	0.51510435	1.565007	-0.046053792	0.5305016
13000.0	3.6882822	0.32379896	1.4243545	2.3450026	0.13044374	0.8107007
14000.0	8.24132	0.6420361	1.5215400	2.905334	0.33050150	1.1072005
15000.0	5.2689675	0.5000169	2.4125702	3.2573300	0.50150307	1.3976471
16000.0	5.4778617	1.2723838	2.5070254	3.6003025	0.60575245	1.6840040
17000.0	7.269916	1.3518884	3.2500105	4.332844	0.8700307	1.9760026
18000.0	5.0588415	1.5002582	3.8343303	4.857758	1.0374005	2.271774
19000.0	9.321871	2.2282494	4.537400	5.4433323	1.2433275	2.5800437
20000.0	12.784588	2.5455245	4.835321	5.8033347	1.4332054	2.898786
21000.0	11.848783	2.8730524	5.300043	6.323203	1.6267253	3.171122
22000.0	11.8989843	3.355124	5.553804	7.000003	1.8358857	3.4700004
23000.0	15.538888	3.32388	6.2383564	7.6024032	2.0347885	3.78243
24000.0	18.578831	3.8354198	6.363827	8.340421	2.2334400	4.0334102
25000.0	19.814871	4.154911	7.44484	8.04848	2.4059558	4.4879886
26000.0	17.8856	4.54111	7.4104288	8.230195	2.6105015	4.7210855
27000.0	18.878862	4.811486	8.1892	8.250122	2.810004	5.1180153
28000.0	18.188017	5.2185852	8.881641	10.285855	3.015488	5.4500085
29000.0	22.127876	5.580124	8.180188	10.820124	3.2212885	5.8815811
30000.0	21.147874	5.5058887	8.810181	11.355435	3.42855	6.1811478
31000.0	22.159854	5.282045	10.080101	11.885852	3.6308003	6.307457
32000.0	26.171592	5.812185	10.552338	12.420041	3.8485258	6.6861358
33000.0	28.185156	6.5353394	11.030135	12.558055	4.0013473	6.9861050
34000.0	25.184238	7.338488	11.503725	13.463655	4.235702	7.33818
35000.0	25.288843	7.3005886	11.574853	14.035783	4.403030	7.6715673
36000.0	27.28536	8.00307	12.448855	14.555557	4.3000543	8.034527
37000.0	28.287558	8.437727	12.518156	15.00344	4.5304488	8.358005
38000.0	23.287636	8.830742	13.38038	15.827355	5.351300	8.704110

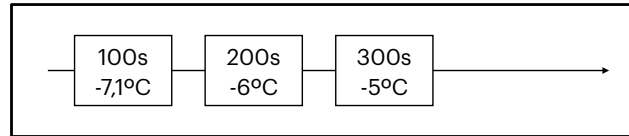
Small Room
With Window (T₁):

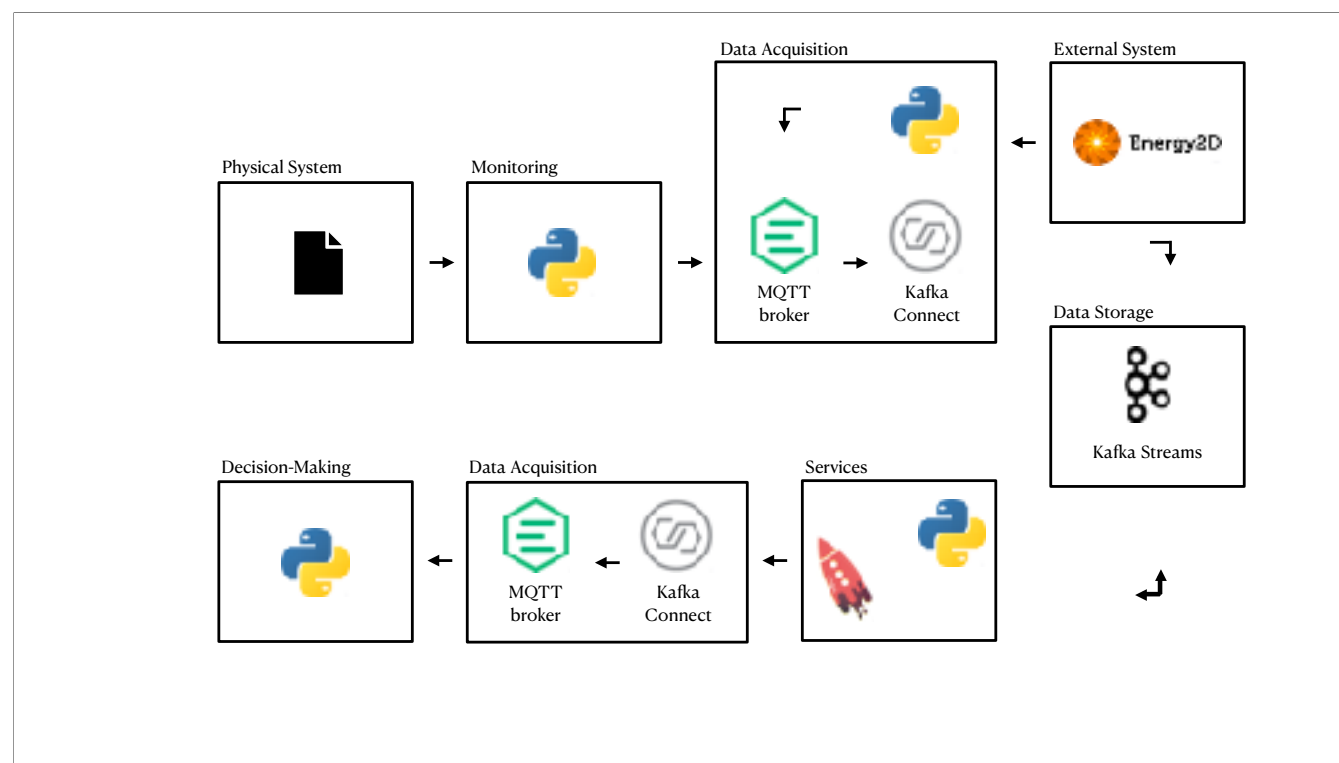


Big Room (T₂):



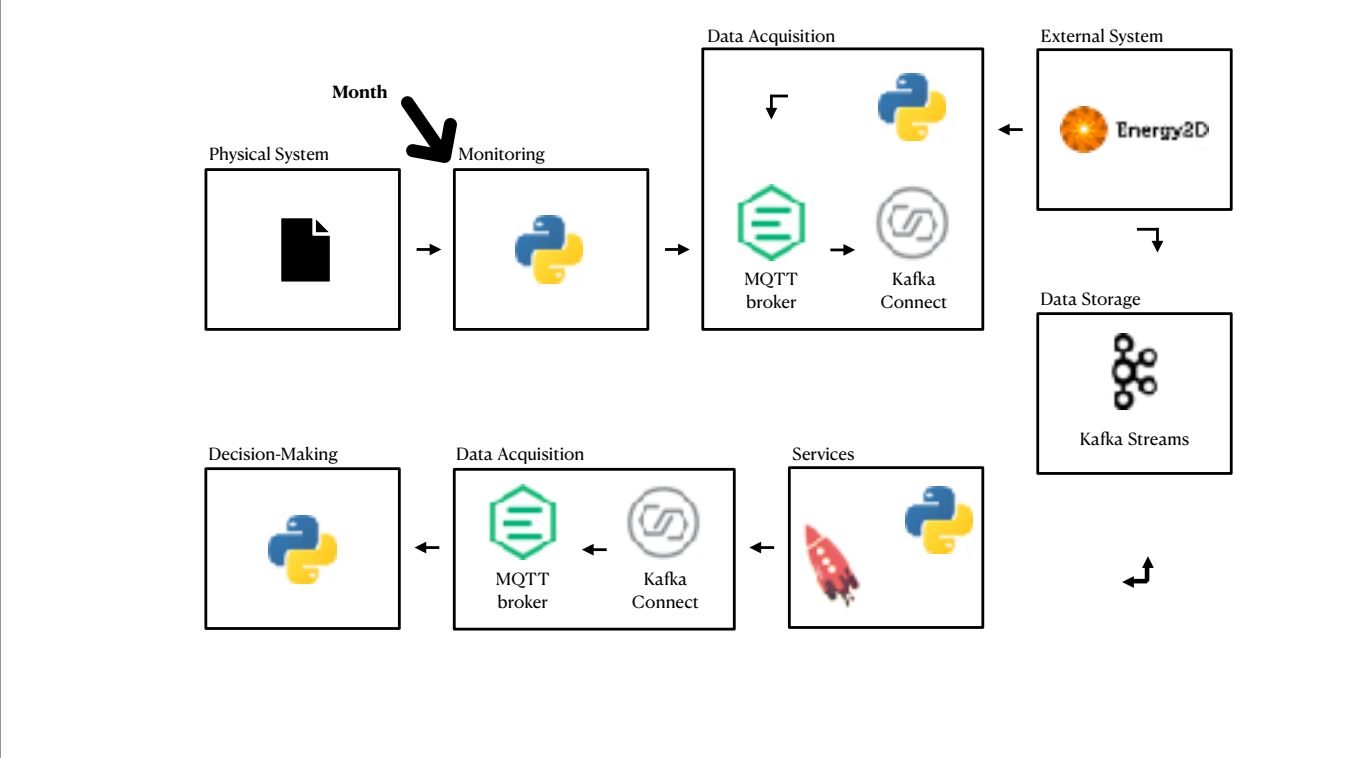
Small Room (T₃):

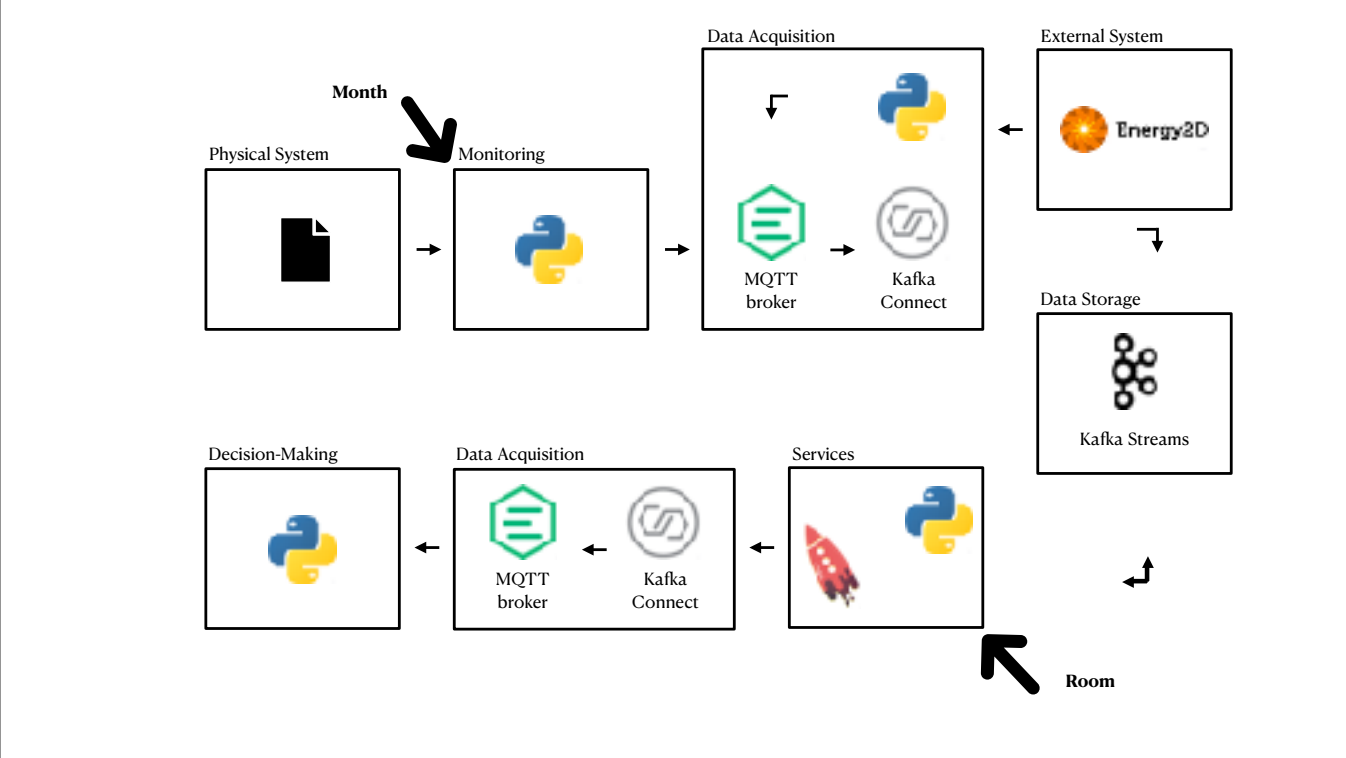




Acrescentamos um novo parametro de execução ao script de monitoramento, que é um número de mês. Assim, ele publica apenas os dados referentes àquele mes do arquivo de entrada.

Também o script de serviço passa a receber como argumento o nome do quarto, assim ele sabe qual dos 3 streams de simulação presentes do Kafka ele deve consultar.

















Evaluation criteria 1

Thermal comfort

	<table><tr><th>Final Temperature (°C)</th></tr><tr><td>19,5</td></tr><tr><td>19,2</td></tr><tr><td>20,5</td></tr></table>	Final Temperature (°C)	19,5	19,2	20,5	—	20°C	=	<table><tr><th>Difference (°C)</th></tr><tr><td>0,5</td></tr><tr><td>0,8</td></tr><tr><td>0,5</td></tr></table>	Difference (°C)	0,5	0,8	0,5
Final Temperature (°C)													
19,5													
19,2													
20,5													
Difference (°C)													
0,5													
0,8													
0,5													













O primeiro critério comparativo é o conforto térmico. A gente então calculou a diferença entre a temperatura estimada atingida pelo sistema às 18 horas e a temperatura ideal, 20°C.

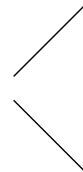
	R1	R2	R3
J			
F			
M			
A			

Após executarmos as 12 iterações, temos os 12 arquivos de saída.













Clica

Retomando, cada um lista os dias do mês em questão, a que horas a ação de ligar o aquecedor foi tomada e a que temperatura o ambiente físico estava.

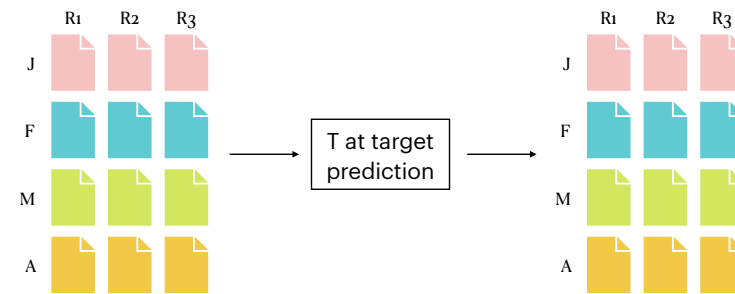
	R1	R2	R3
J			
F			
M			
A			

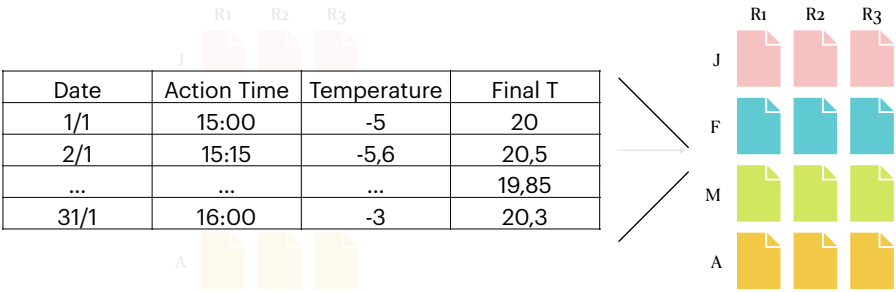


Date	Action Time	Temperature
1/1	15:00	-5
2/1	15:15	-5,6
...
31/1	16:00	-3

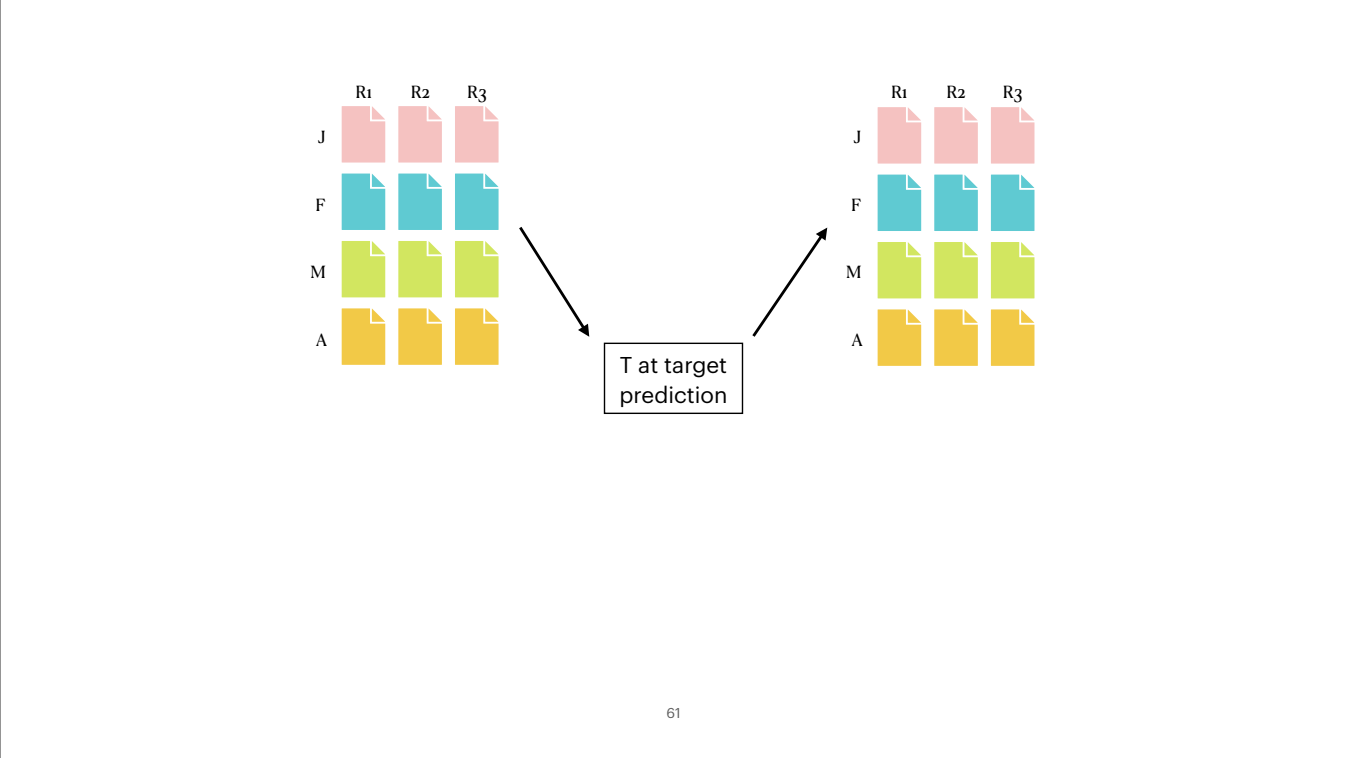
	R1	R2	R3
J			
F			
M			
A			

Depois fizemos um novo script que recebe esses arquivos como entrada e estima a temperatura o sistema chegou as 18 horas. Entao ele tem como saida os mesmos arquivos com uma coluna a mais.

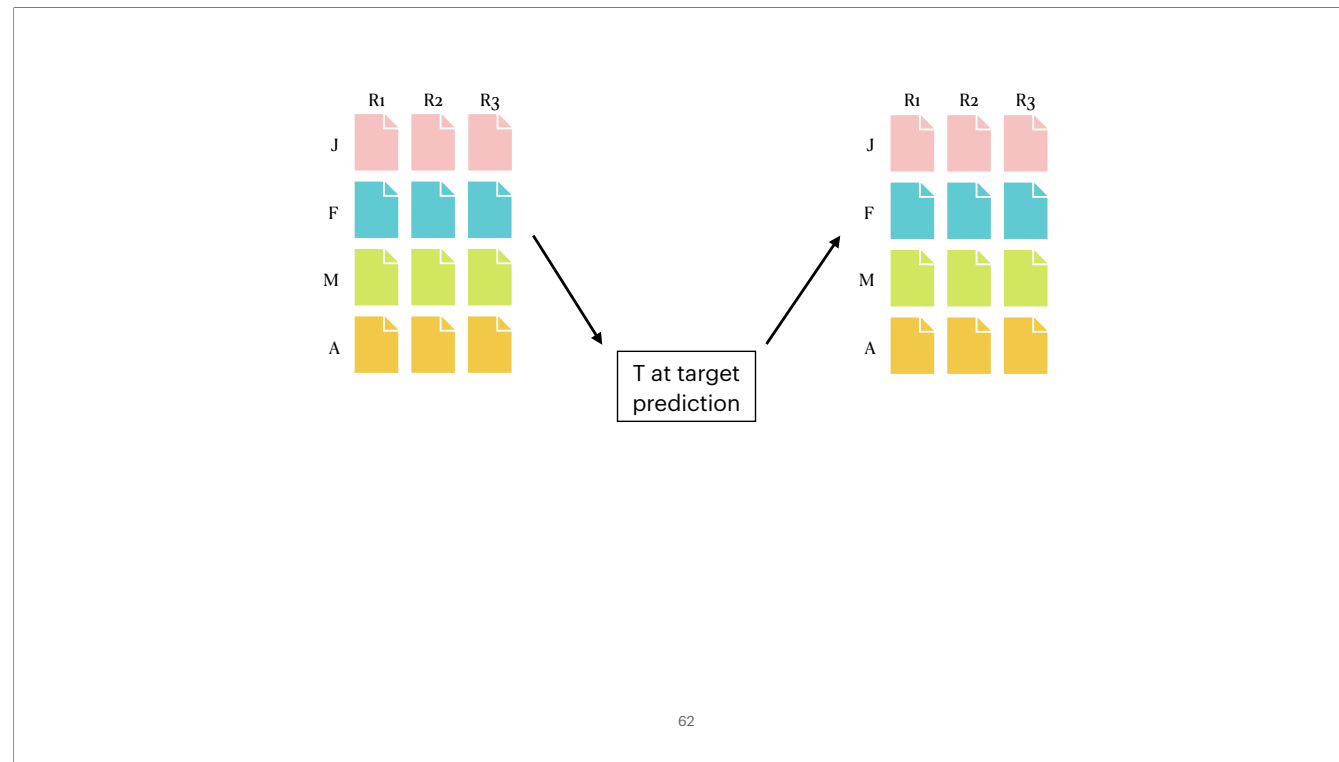








Passa



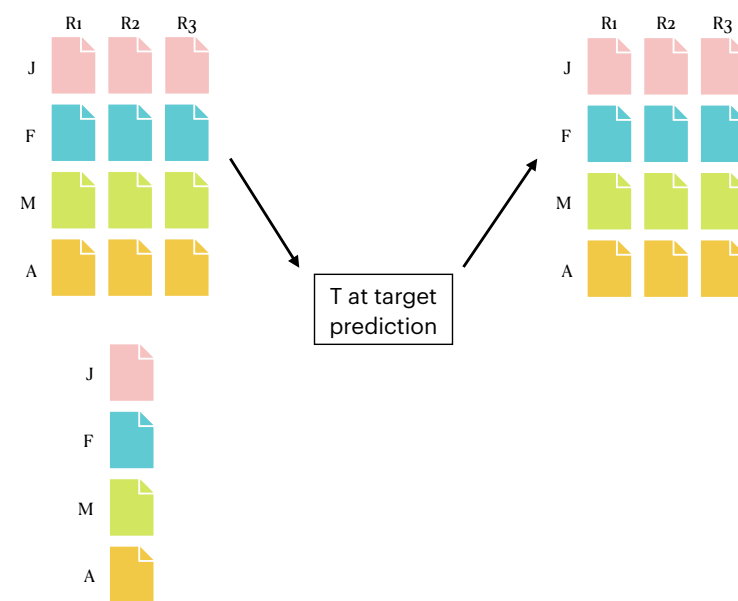
Resolvemos comparar entao esses resultados com os de um sistema que liga todos os aquecedores sempre no mesmo horario. Esse horario fixo varia conforme o mes e é igual pra todos os quartos. Obtivemos esse valor mensal fixo pela media dos horarios em que a acao foi tomada usando o nosso sistema naquele mes.

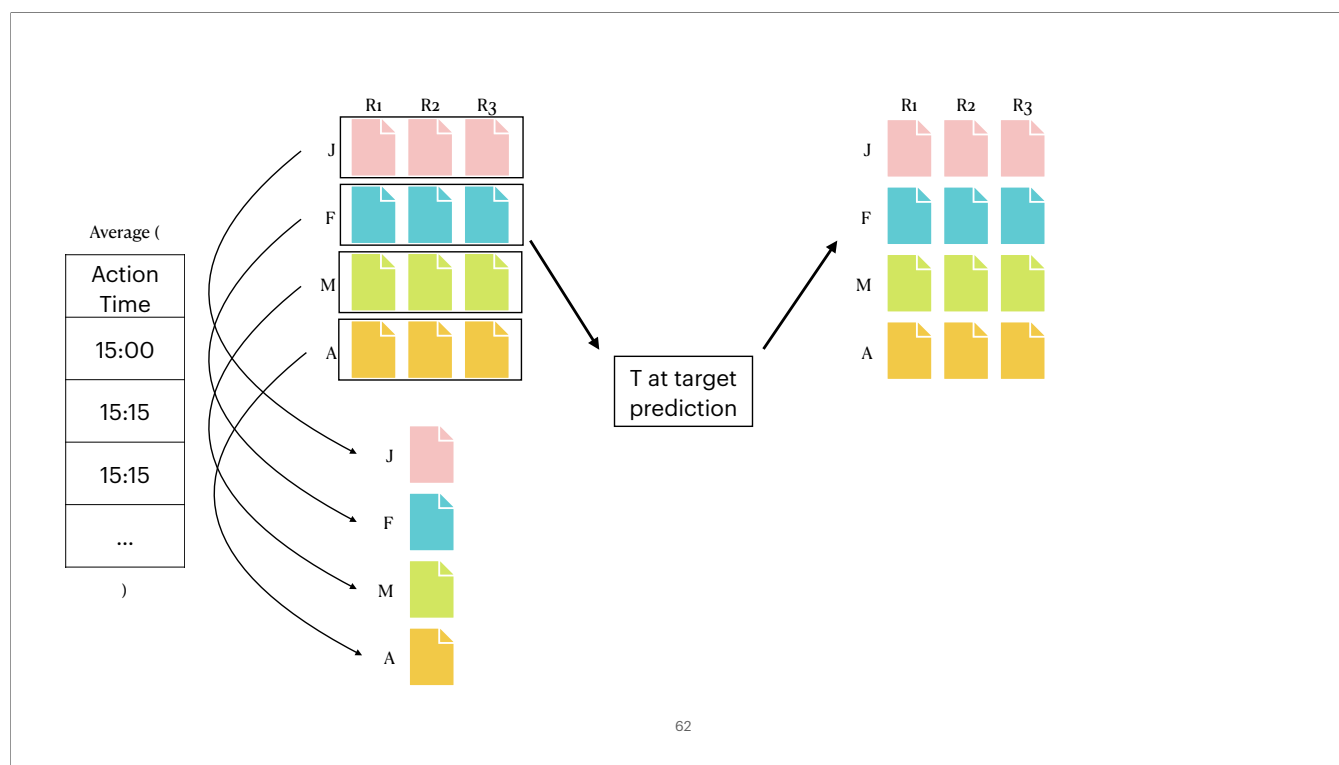
Entao temos quatro arquivos, também com as mesmas tres colunas temperatura, hora da acao e data. A coluna hora da acao tem todas as linhas com o mesmo valor em um mesmo arquivo. Pra determinar a temperatura na qual o sistema fisico estava nesta hora, o script consulta o conjunto de dados que simula nosso sistema fisico (o com as temperaturas de stuttgart).

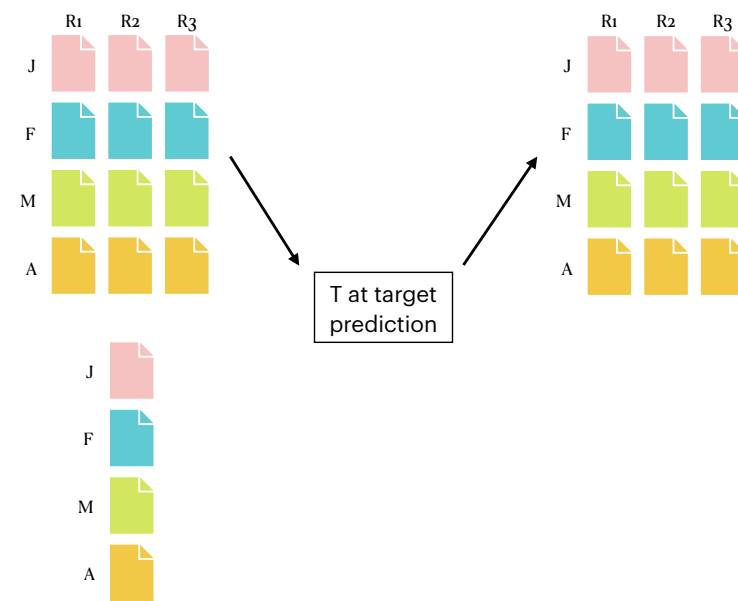
<clica>

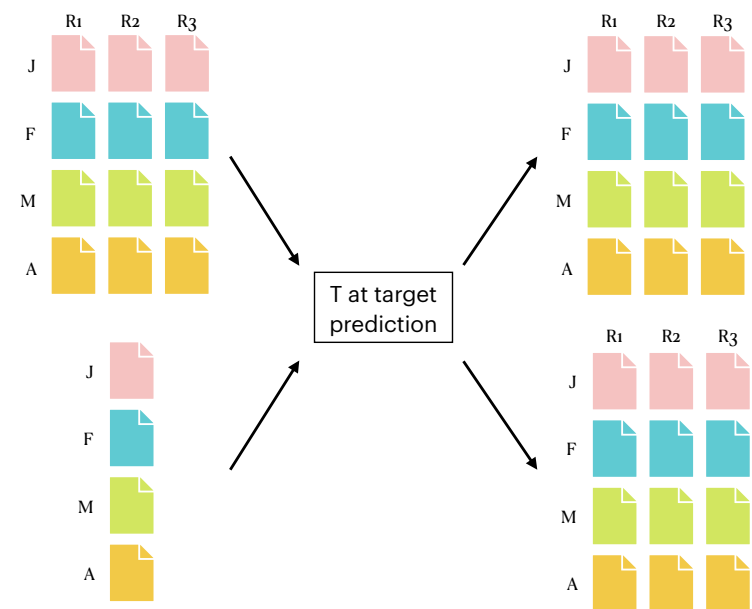
Entao, novamente, esses arquivos passam pelo script que determina a temperatura final do sistema.

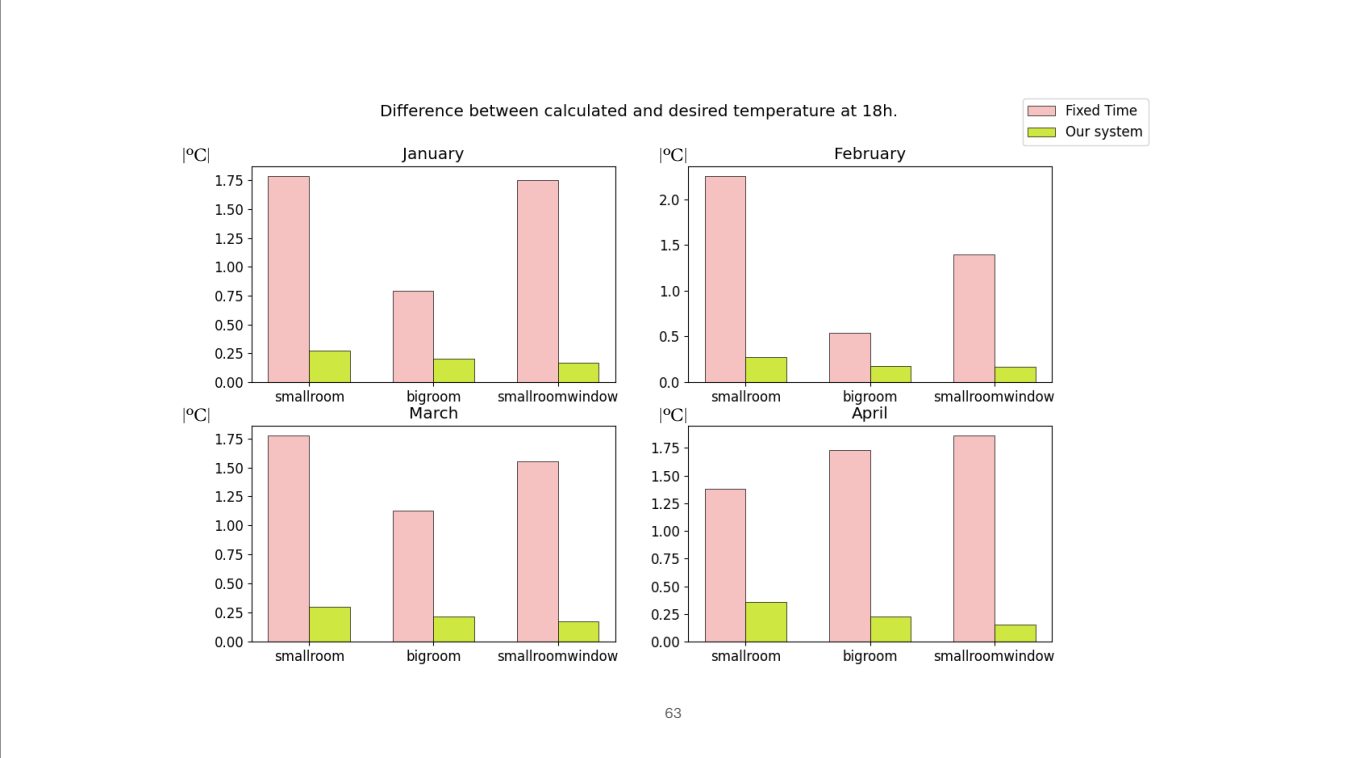
Entao comparamos os dois sistemas a respeito do conforto termico e consumo de energia.











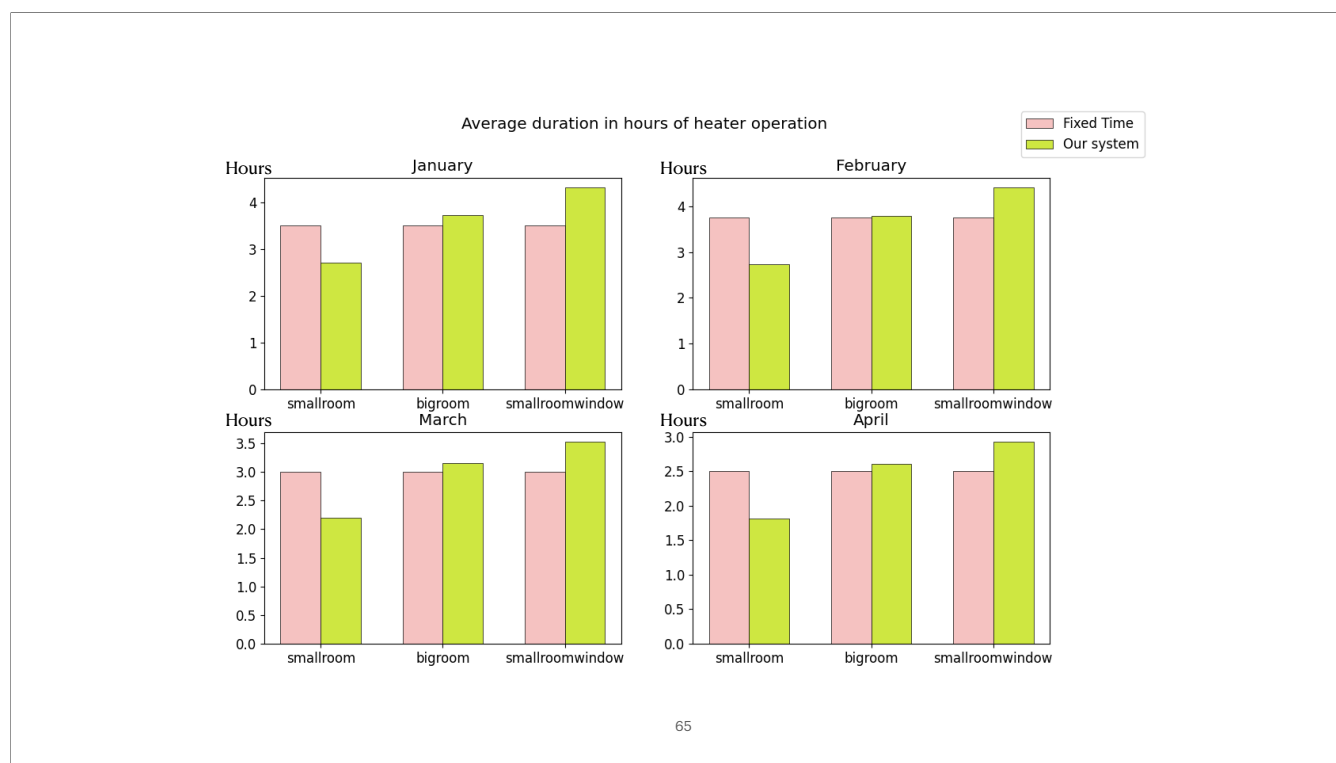
Fizemos então a médias mensais dessas diferenças

Evaluation criteria 2

Energy consumption

	<table><tr><th>Action Time</th></tr><tr><td>15:00</td></tr><tr><td>16:00</td></tr><tr><td>15:15</td></tr></table>	Action Time	15:00	16:00	15:15	—	18:00	=	<table><tr><th>Difference (h)</th></tr><tr><td>3</td></tr><tr><td>2</td></tr><tr><td>2,75</td></tr></table>	Difference (h)	3	2	2,75
Action Time													
15:00													
16:00													
15:15													
Difference (h)													
3													
2													
2,75													

O segundo critério comparativo é o consumo de energia, em que medimos o tempo que o aquecedor permaneceu ligado desde a hora em que a ação foi tomada até às 18 horas.



Então novamente fazemos as médias mensais. Não houve muita diferença entre nosso sistema e o de tempo fixo.

Geracao da hora é a media dos tempos do nosso -> em média é igual.

Como o quarto pequeno demora menos pra aquecer, é ligado depois da média, entao o nosso sistema ganha. Como o de janela a hora ideal é depois da media, o nosso sistema perde. No do meio é praticamente igual.

Summary

1. Introduction
2. Use Case 1
 1. Introduction
 2. Architecture
 3. Demo
3. Use Case 2
 1. Introduction
 2. Architecture
 3. Evaluation
4. **Limitations**
5. Conclusion

Partindo agora pro caso de uso 2

Limitations

- No real system
- Assistant Software Energy 2D
 - 2D
- No people's interactions

Summary

1. Introduction
2. Use Case 1
 1. Introduction
 2. Architecture
 3. Demo
3. Use Case 2
 1. Introduction
 2. Architecture
 3. Evaluation
4. Limitations
5. **Conclusion**

Conclusion

- Demonstrated use cases for DT's two roles
 - Interrogative
 - Visualization and analysis of power consumption
 - Predictive
 - 85% better thermal comfort

Obrigada!