Enabling Personalized Context-Aware Applications via the Web of Things

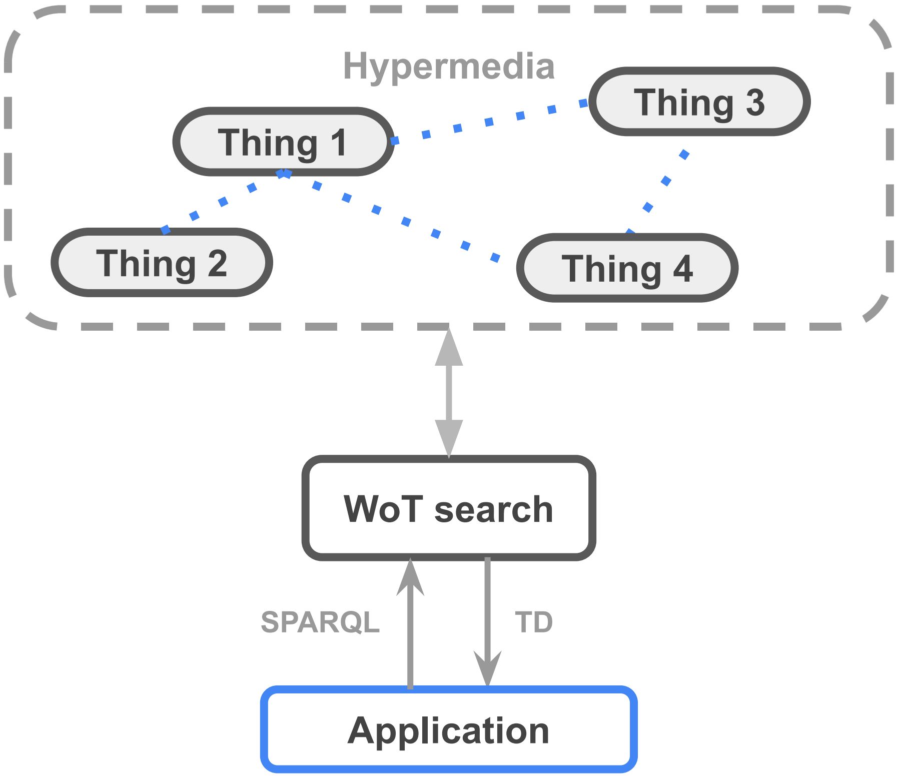
Semester Project: Final Presentation

Massimo Albarello

18/02/22

Supervisors: Dr. Kimberly Garcia, Dr. Andres Gomez, Ganesh Ramanathan
Professor: Pr. Dr. Simon Mayer

Enabler



Web of Things:

- Thing Description
- hypermedia environments
- WoT search engine

Applications:

- personalized
- context-aware

Application: smart light

Observe

- discover users in the room
- acquire data and associate it to the correct user

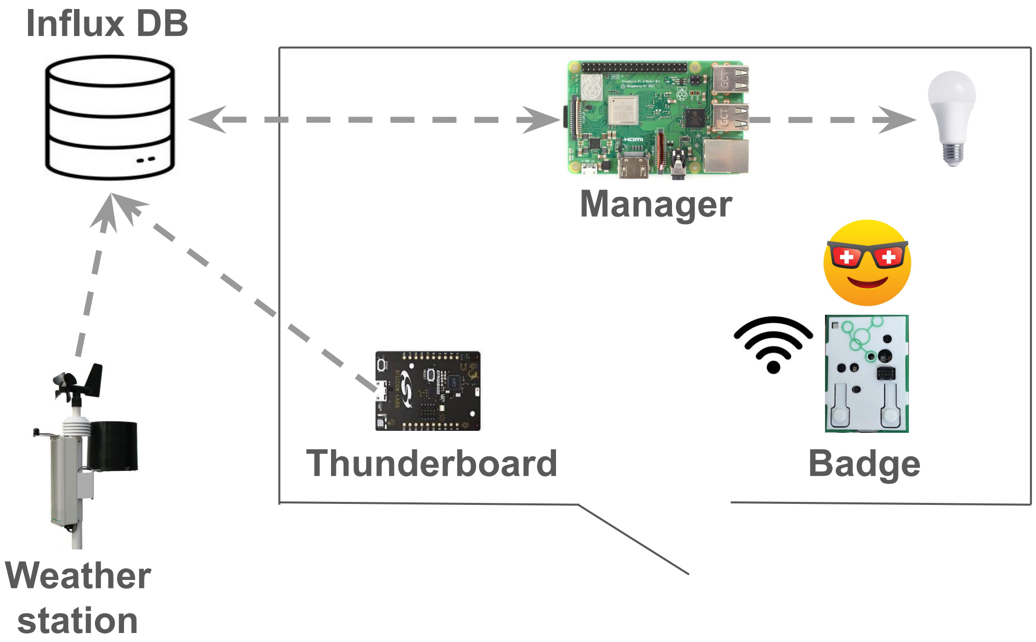
Learn

- once enough data is collected, train a machine learning model

Predict

- make predictions about the state of the light based on current data
- set state of the light according to the prediction

Room setup



Manager: runs the smart light app

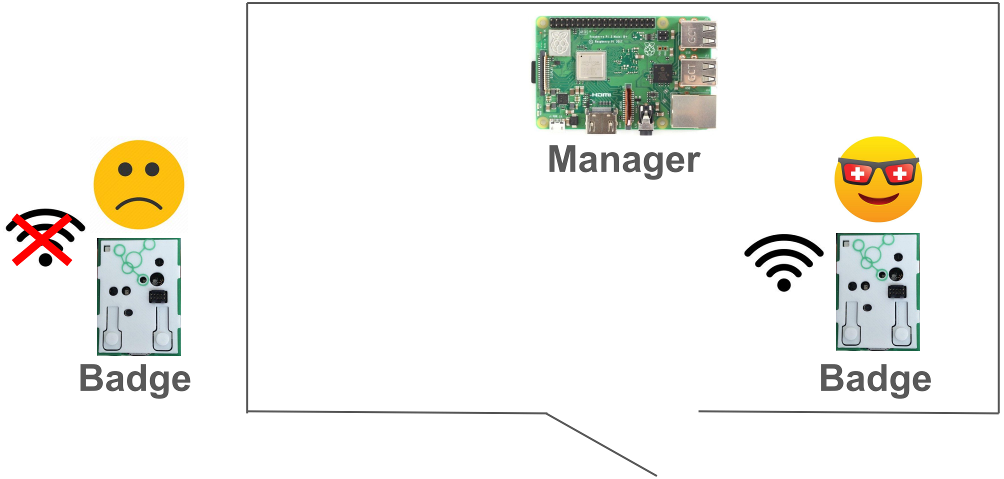
Badge: identifies each user

Thunderboard: measures in-room brightness, temperature, humidity, …

Weather station: measures external light, sun elevation, wind speed …

Influx DB: stores data acquired from sensors

Discover users in the room

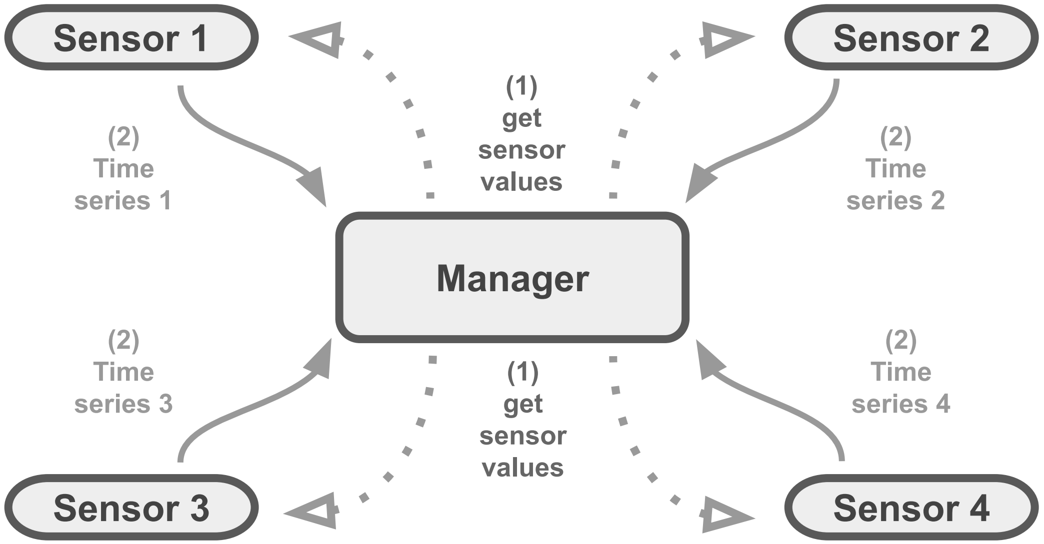


- badge periodically broadcasts via BLE
- RSSI value used to estimate whether user is in the room or not

+ simple approach
+ correctly discriminate users inside/outside (most of the times)

- coarse granularity localization
- users have to constantly wear the badge

Data acquisition



Snapshot

{

Time series 1

Time series 2

...

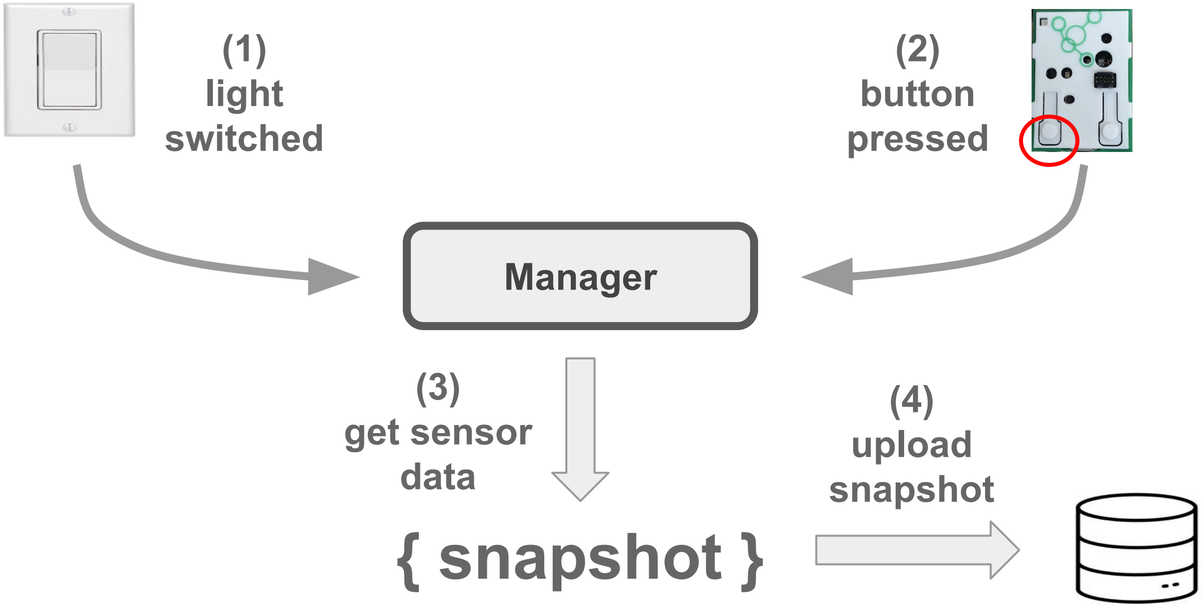
Metadata

Label

}

Time series: values measured in the last 20 minutes (1 value per minute)
Metadata: time and day of the week when the snapshot was acquired and whether someone was in the room
Label: current state of the light (on/off)

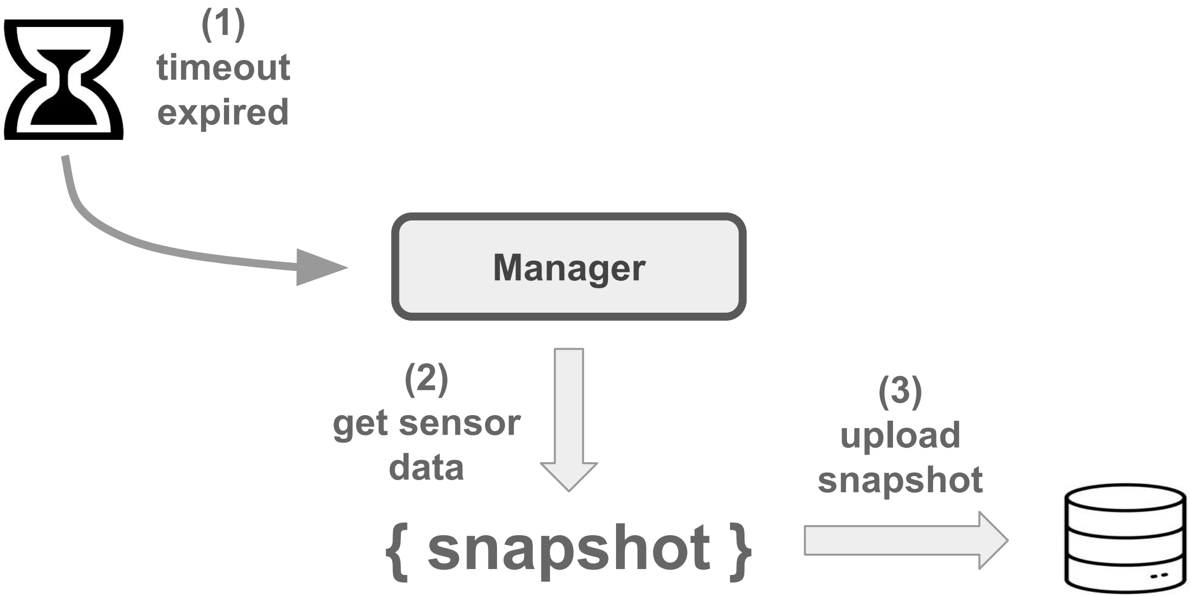
Acquire personal preferences



+ snapshots grouped according to the user who switched the light

- few snapshots per user collected daily
- no data collected when light is not switched
- count on the user to press the button

Uniform data acquisition



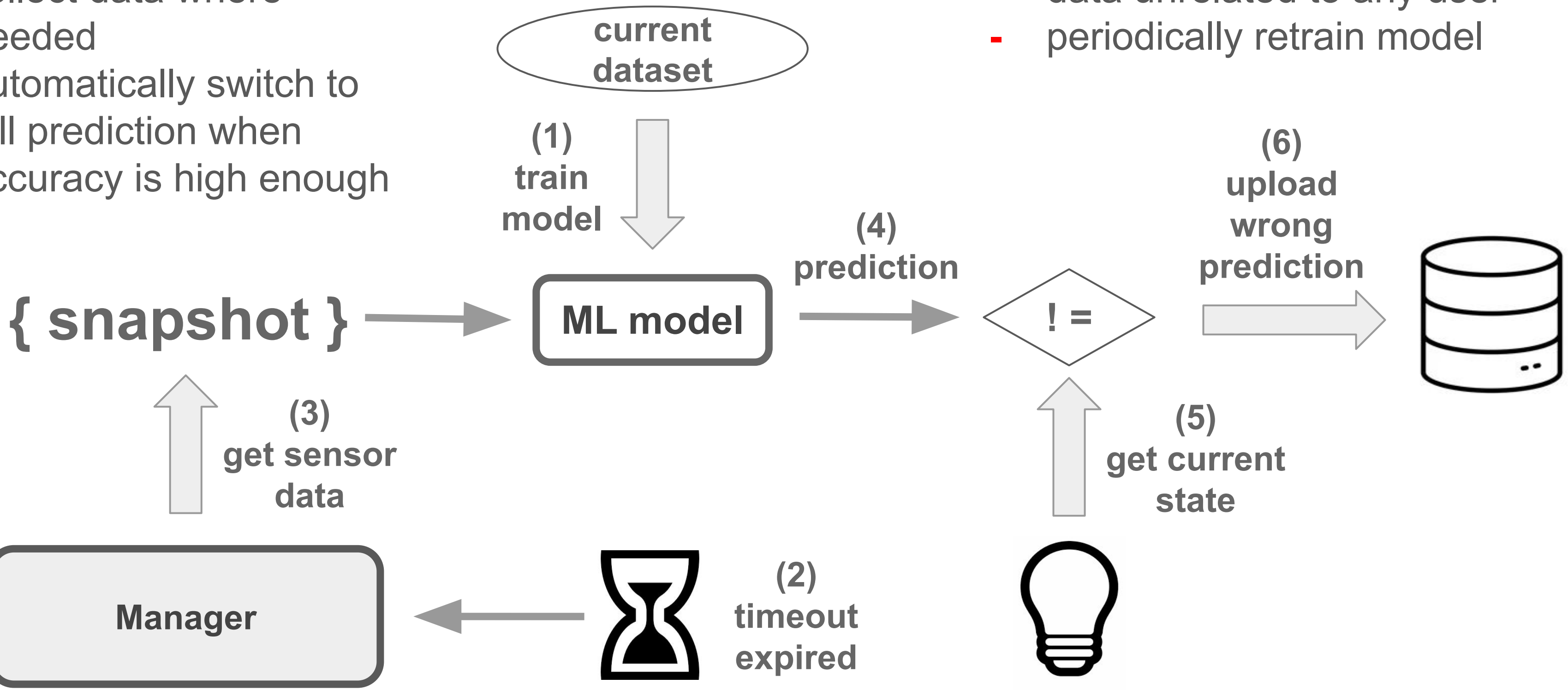
+ reduce time of data acquisition phase
+ collect data in all circumstances

- data unrelated to any user
- does not regulate acquisition according to uncertainty

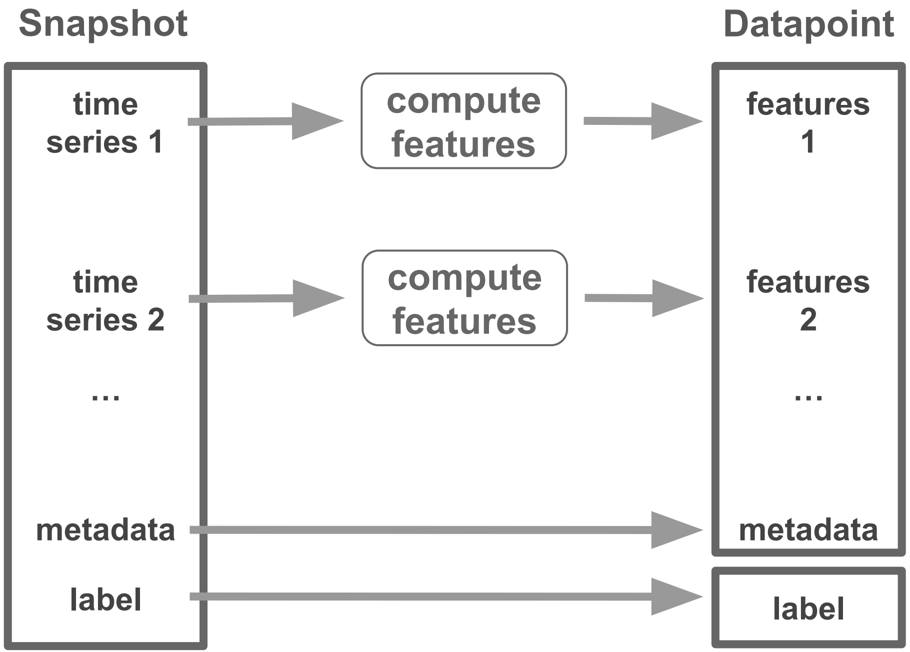
Uncertainty-based acquisition

+ collect data where needed
+ automatically switch to full prediction when accuracy is high enough

- data unrelated to any user
- periodically retrain model



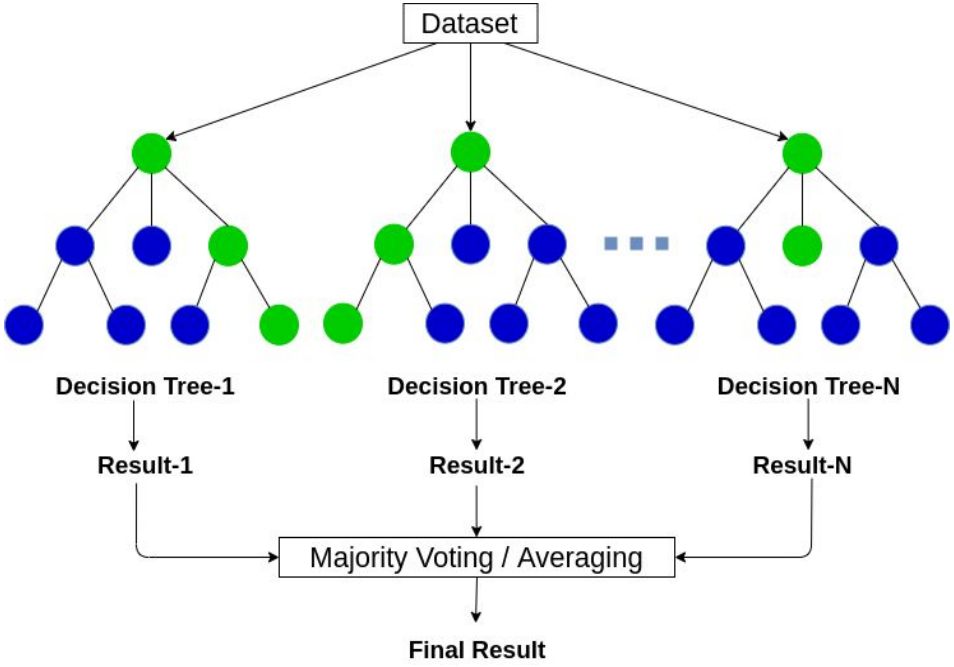
Preprocessing



Features (for each time series):

- last sensor measurement
- standard deviation
- average of the last 3 values
- max variation of the last 3 values
- average of the remaining values
- max variation of the remaining values

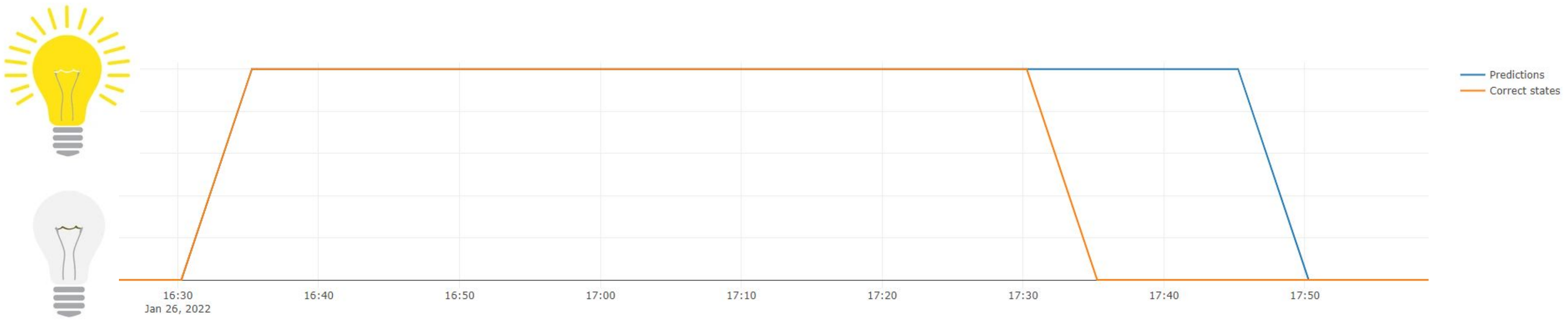
Random Forest Classifier



- reduces prediction variance
- can be trained on a Raspberry Pi
- requires little data

Breiman, L. Random Forests. *Machine Learning* 45, 5–32 (2001). https://doi.org/10.1023/A:1010933404324

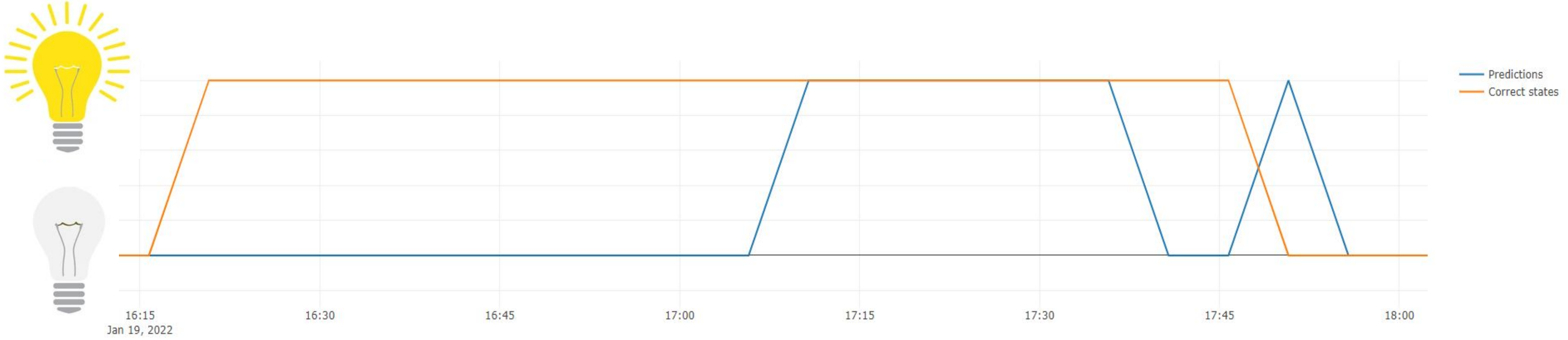
Evaluation (1)

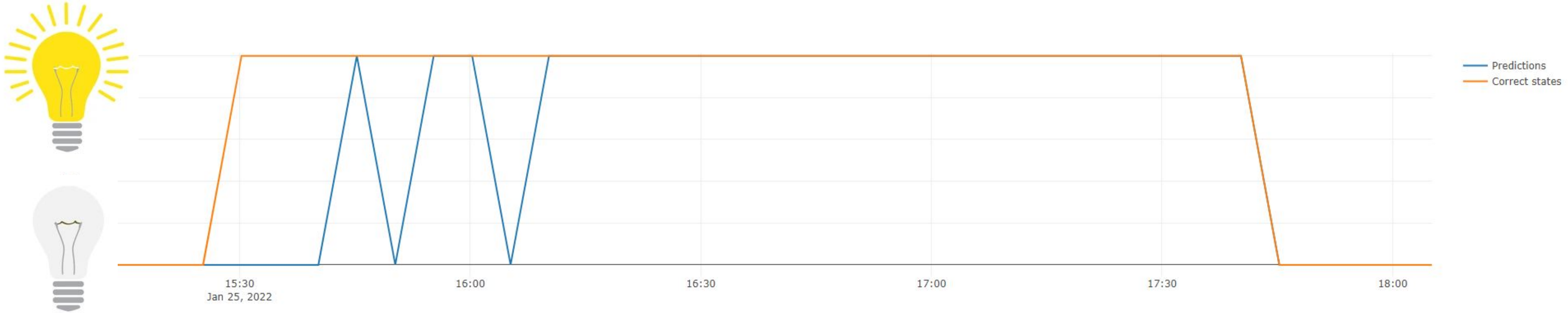


Promising approach:

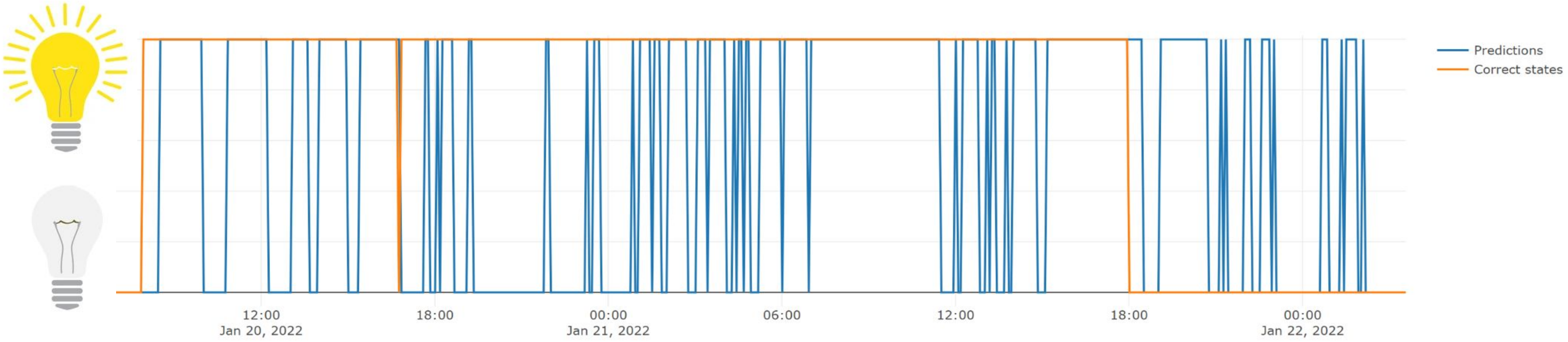
- prediction accuracy around 90%
- 15 days of correct data acquisition
- no optimization of features selection and machine learning model

Evaluation (2)



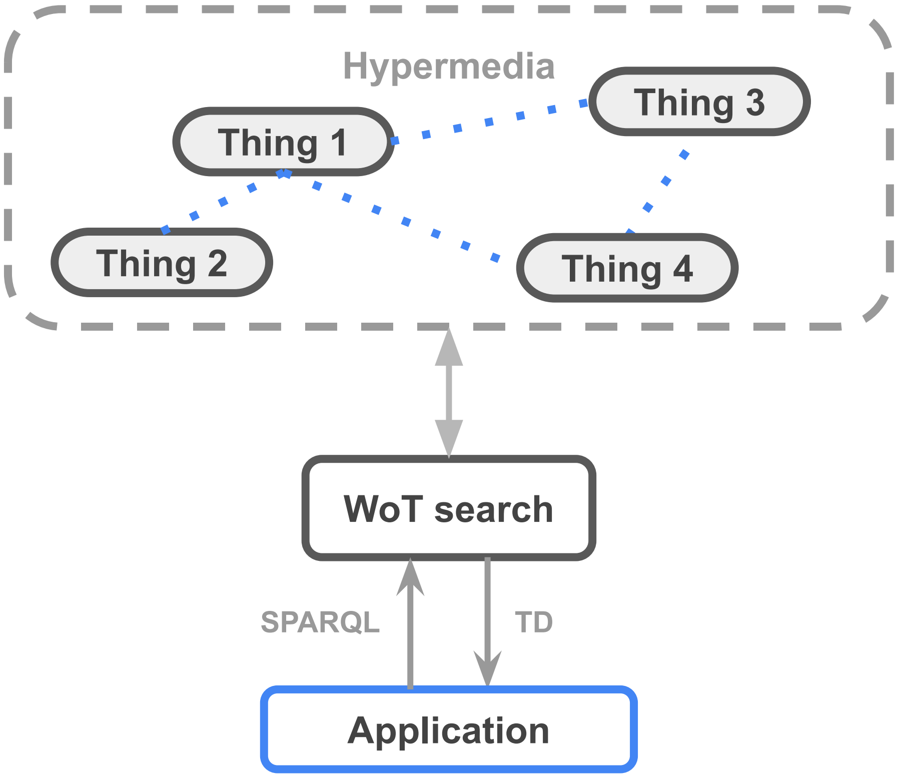


Evaluation (3)



Model cannot discriminate outlier conditions

Summary

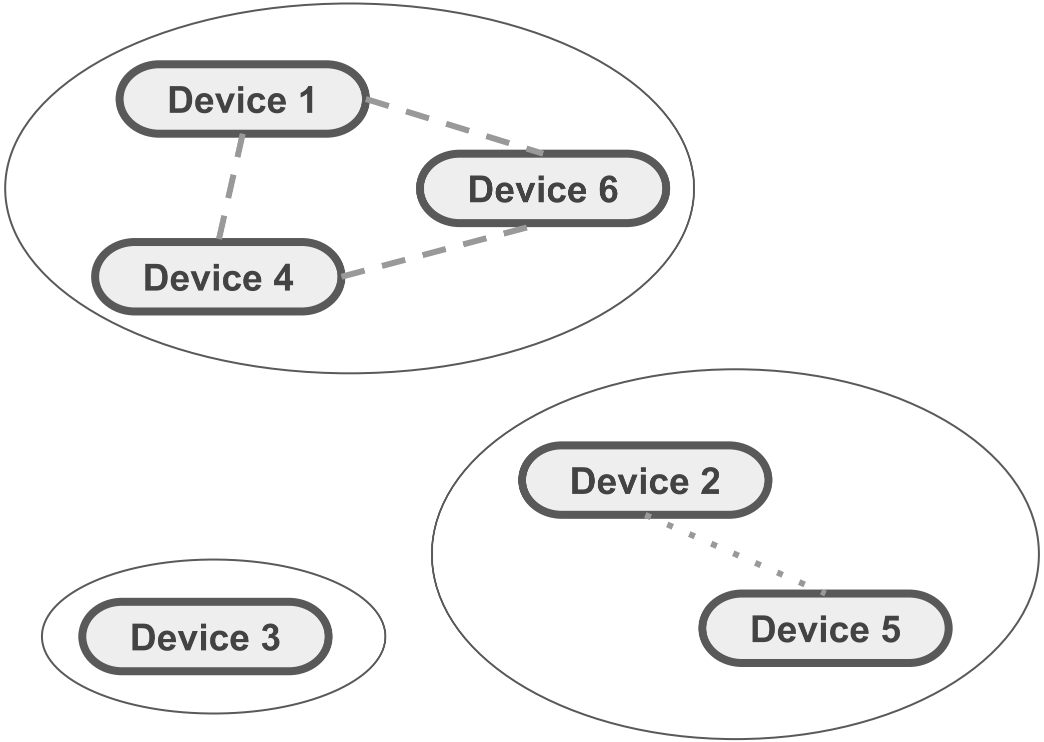


Personalized applications:

- understand the user context
- shape the environment according to learned preferences.

Many applications that can be developed following this approach (smart blinds, smart AC, …)

Problem



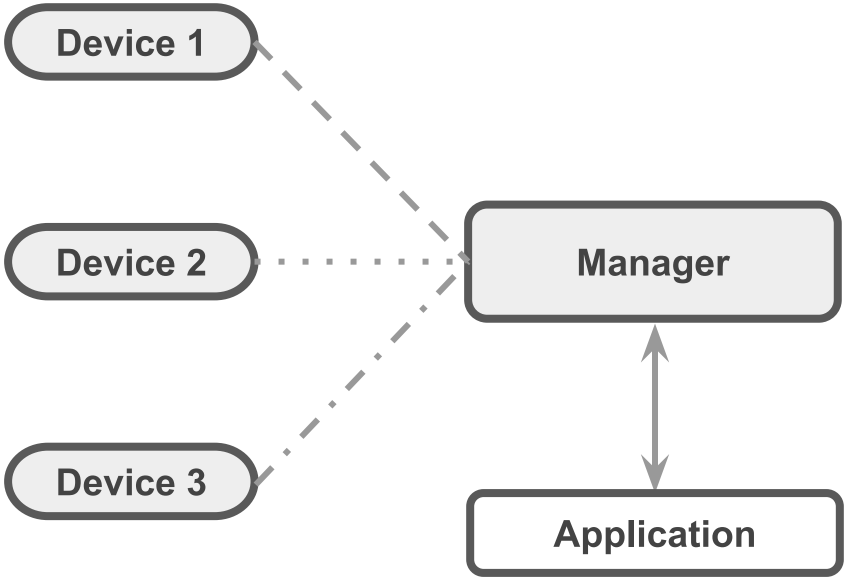
IoT devices:

- use different application protocols
- data is siloed

Applications:

- must interface via different application protocols
- cannot benefit from all data

Solution ?



Advantages:

- manager “talks” the different application protocols
- application interfaces only with manager

Disadvantages:

- not flexible if a device is no longer available
- not scalable to many devices using different protocols

Application: not a smart light

Application: smart light

Training

get snapshots

{

Snapshot 1: { … }

Snapshot 2: { … }

…

}

Snapshot

{

Time series 1

Time series 2

…

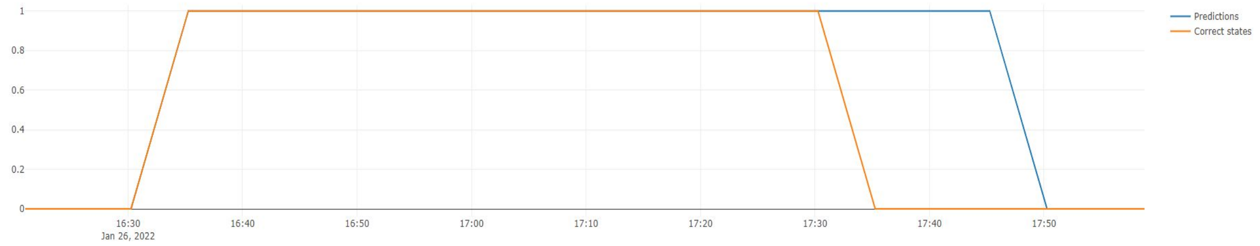
Metadata

Label

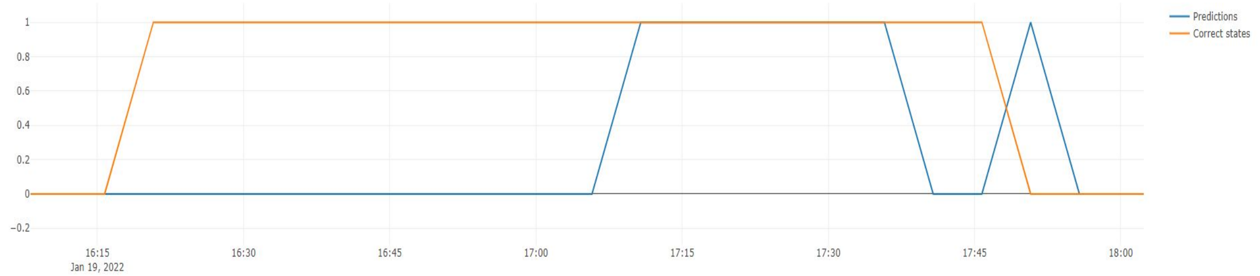
}

Time series: sensor values measured in the last 20 minutes (1 value per minute)
Metadata: time and day of the week when the snapshot was acquired and whether someone was in the room
Label: current state of the light

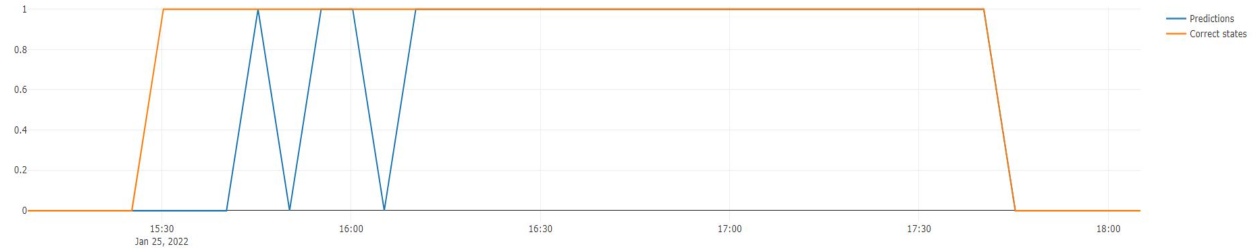
Set the state of the light



correct prediction



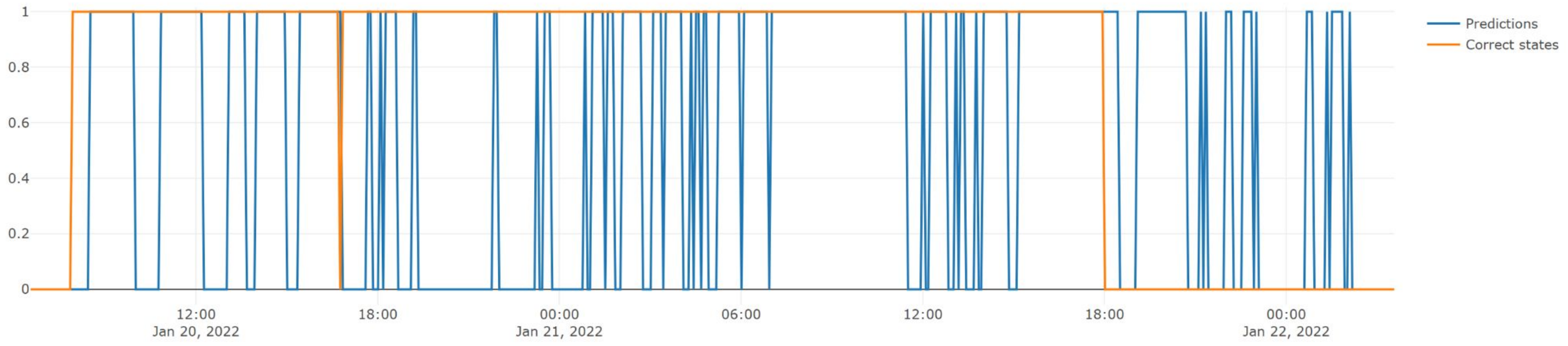
shifted prediction



oscillating prediction

Conclusion

- promising approach: prediction accuracy around 90% even with just 15 days of correct data acquisition and no optimization of features selection and machine learning model
- more data has to be acquired to prevent oscillations, imposing rules may momentarily fix problems but makes the system less flexible
- main concerns due to use outside of lab environment (cannot count on the user to always wear the badge, model cannot discriminate outlier conditions)



Smart light ?

Smart light !

Steps to a truly smart light

Observe

- find users in the room
- acquire data and associate it to the correct user

Learn

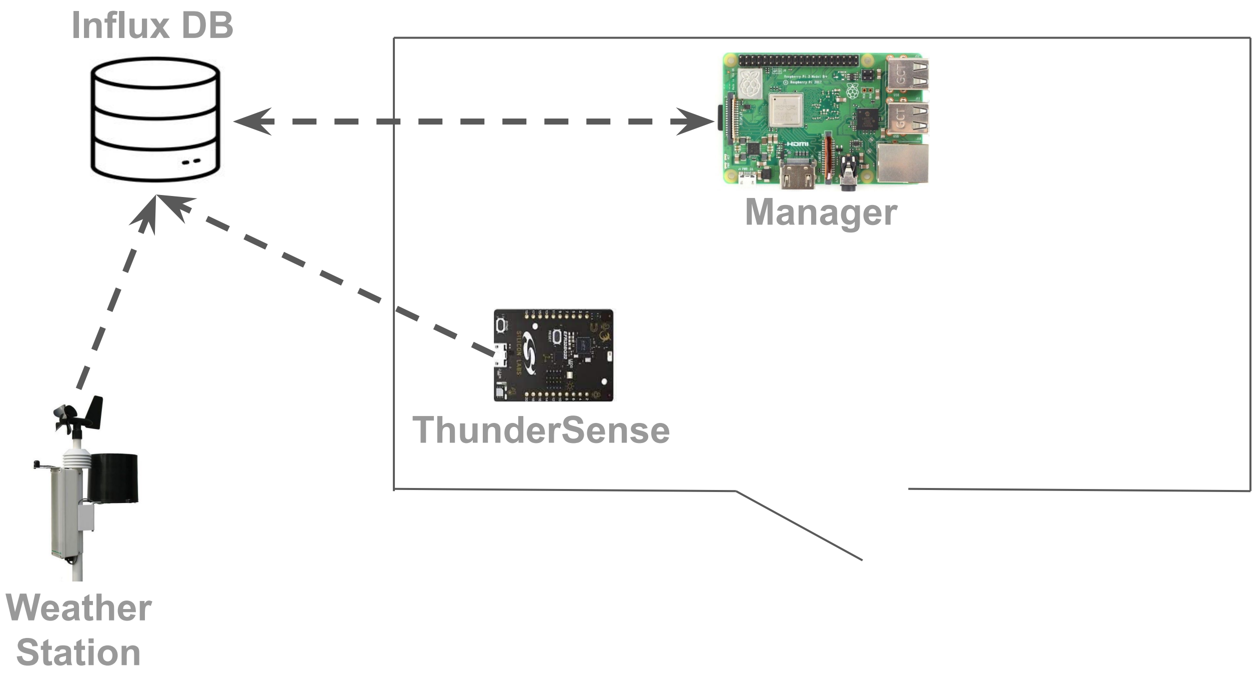
- once enough data is collected, train a machine learning model

Predict

- make predictions about the state of the light based on current data
- set state of the light according to the prediction

Observe

Room hardware setup



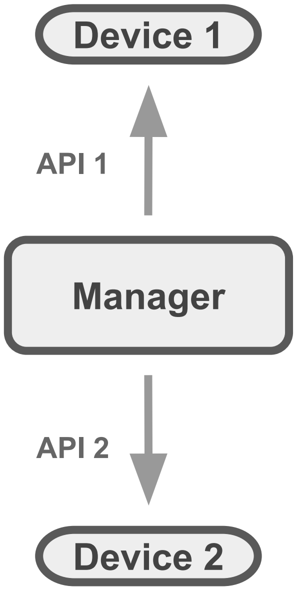
Learn

Predict

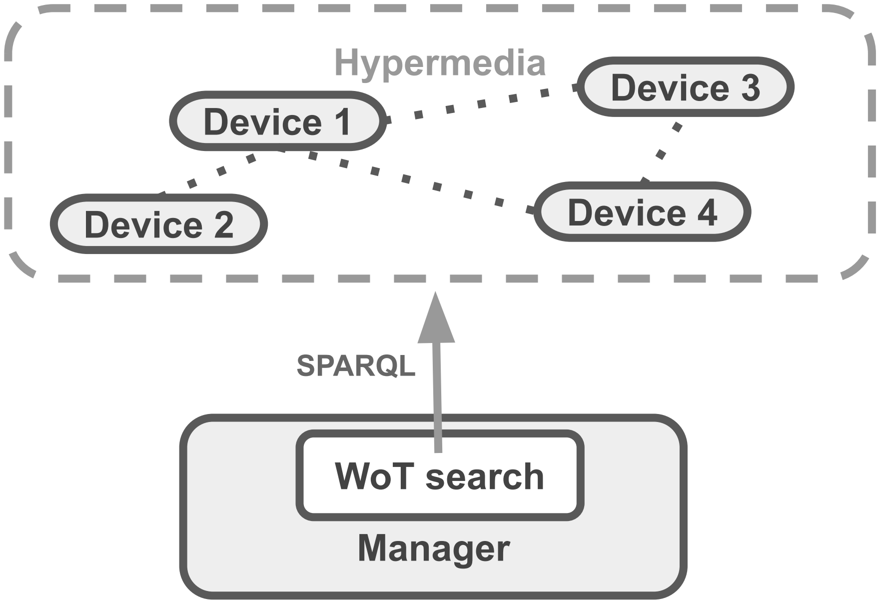
Evaluation

Future work

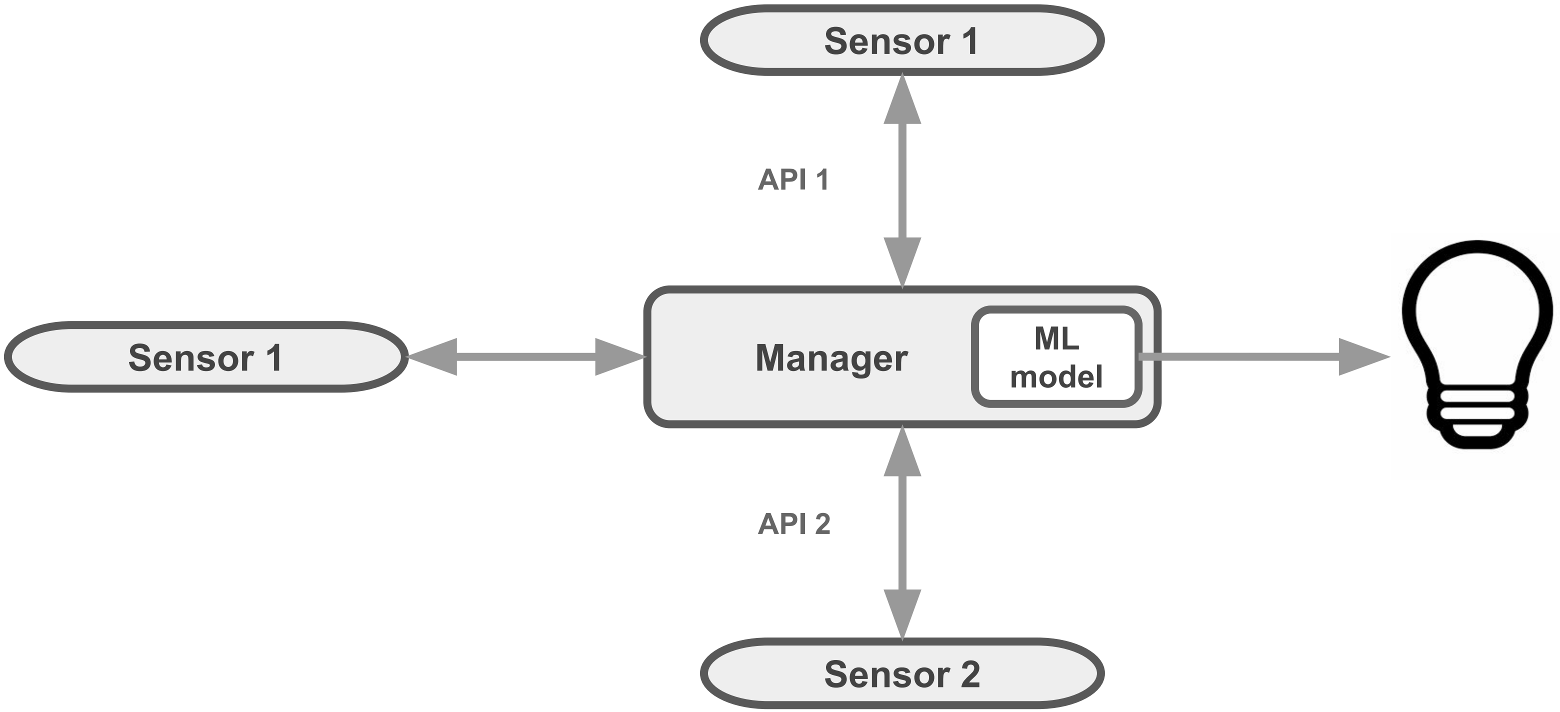
Today



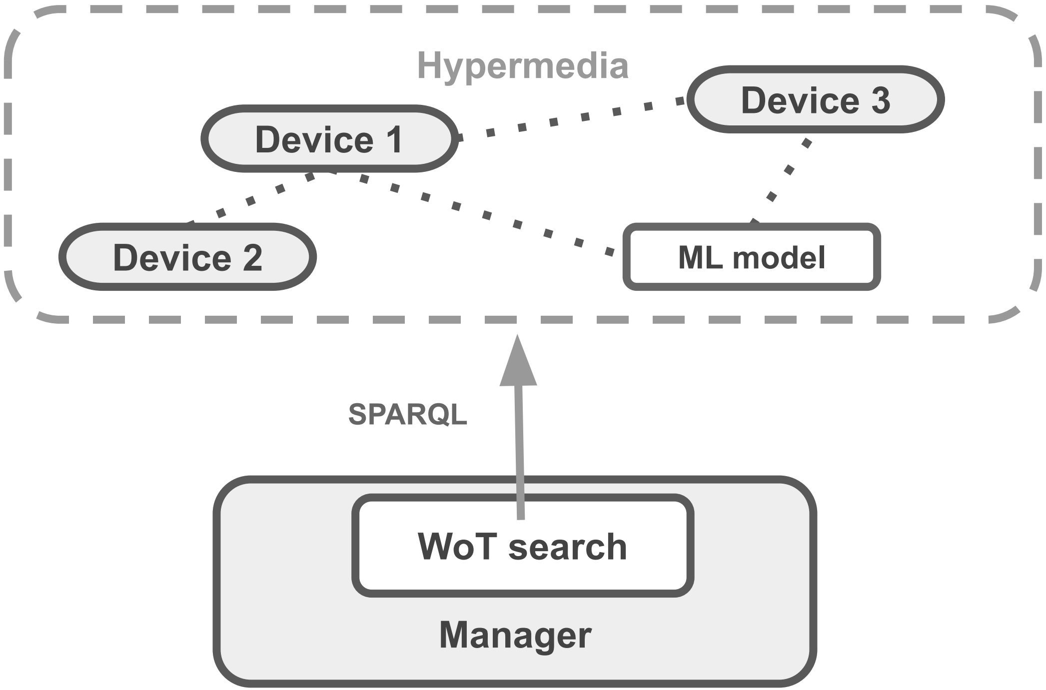
Tomorrow



Today



Tomorrow


Hypermedia
Device 1
Device 3
Device 2
ML model
SPARQL
WoT search
Manager