

Project
STARK



Dev's Kitchen idea

During Dev's Kitchen a team consisting of several students, supported by experienced mentors, is working together to achieve a set goal.

It's all about executing a „recipe” for a Proof of Concept-level project in just under 24 hours and present it to the wide audience.

Problem

Unexpected car battery discharge, especially in the winter.





Solution

Give drivers a solution that uses the computing cloud and IoT (Internet of Things) which will:

- Anticipate potential issues with a car's morning ignition.
- Alert the driver early on.

Existing solutions

- Costly.
- Only ever used during direct diagnostics at a car service workshop.
- No real time analysis of the car battery; reacting accordingly in the nick of time is impossible.
- They require the user's time and attention to analyze.



STARK

- Easy installation, whether it's by the driver or the battery producer („Connected car battery”).
- Real time car battery monitoring (instant anomaly detection).
- Instantaneously informs the driver and nearest workshop via text message.

Business model

Two sides of monetization:

- Workshop - driver.
- Producer - driver.



Business model

workshop and producer

Additional „safe battery” service pack- car battery monitoring and exchange in the right moment.

Gathering and analysis of battery performance data for the producer.



Business model

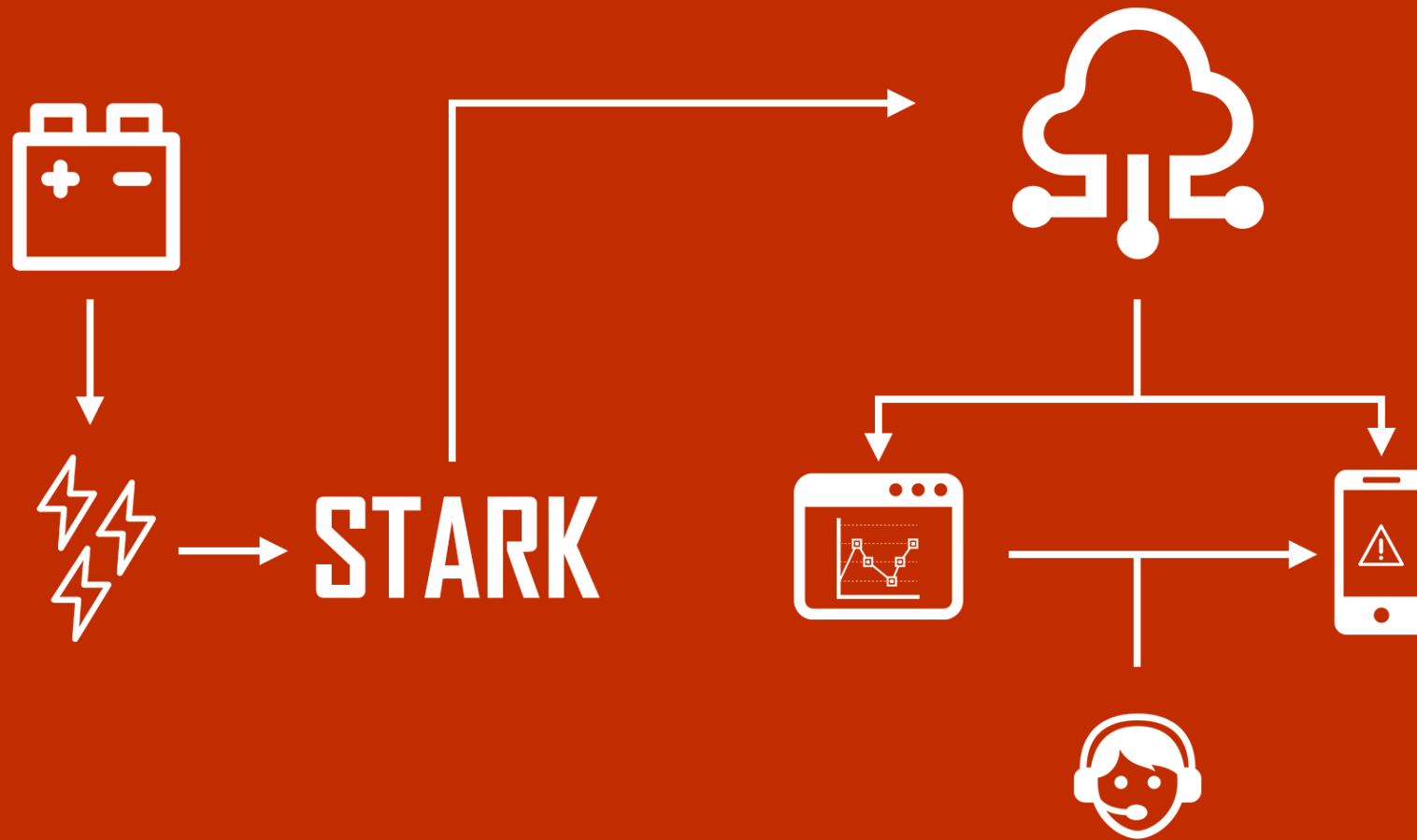
B2B and B2C monetization

- Partnership with the battery producer (pre-installed device).
- Retail sales through workshops and car services.
- Wholesale through distributors and wholesalers.
- Service pack sales by workshops and car services.

Project

STARK

Functionality and technology



Monitoring the battery voltage

The first and most basic functionality of the STARK system is to monitor the car battery voltage. Unfortunately, the few existing microcontrollers are based on 12V logic, while the most popular solutions work in 5V logic, commonly used in many ATMega controllers or 3V3, popular in ARM processors, also in Netduino. Irrespective of the type of microcontroller used, it's most likely to deliver several analog inputs, which will enable easier voltage readings. There is an emerging problem with acquiring the full range of input voltages compatible with the processor used. It can be easily done by using a voltage divider, based on passive elements.

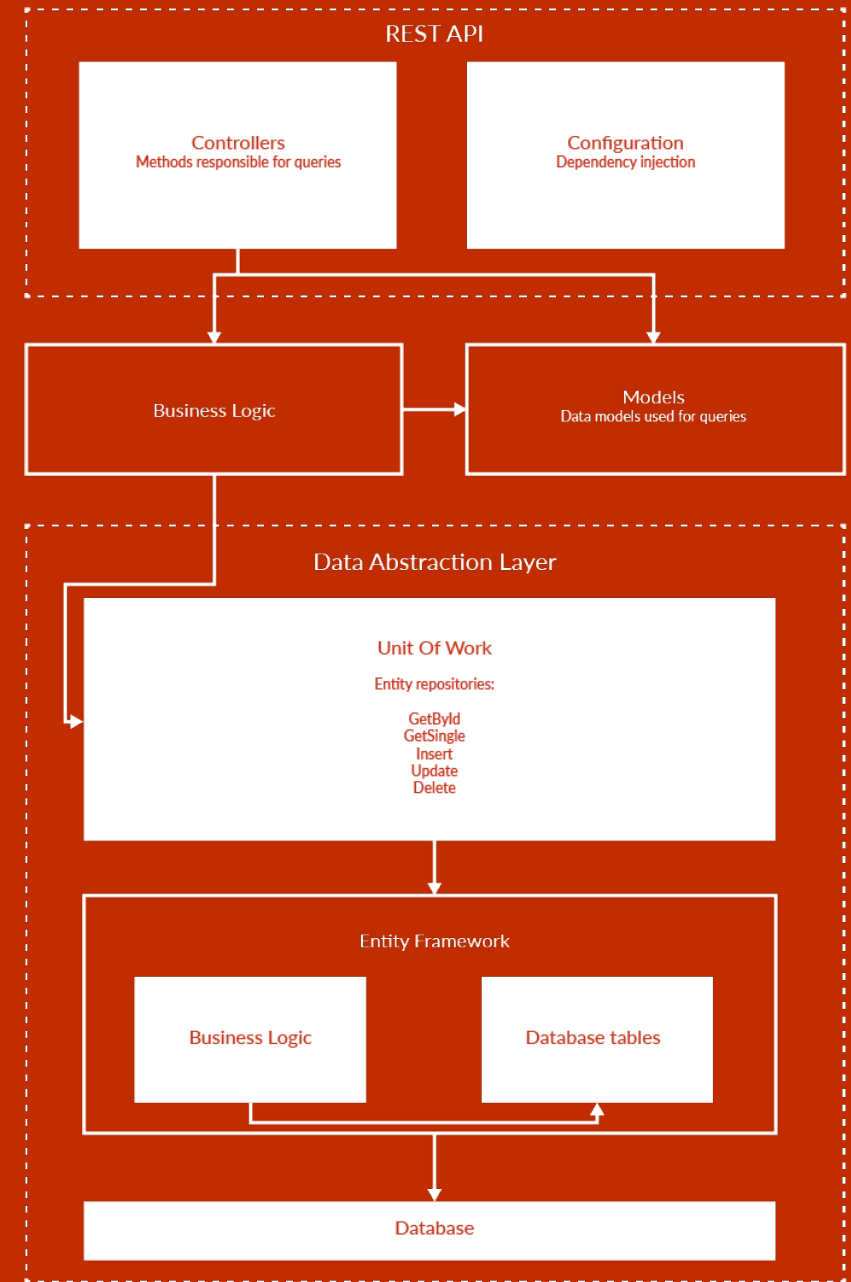
Sending the data to cloud server

The next step of building the system is to establish communication with the computing cloud. It will enable thorough analysis of the acquired data and the adding of additional functionalities. It will require the device to be connected to the internet, which can be provided by a GSM modem that will support packet communication using for example GPRS or UMTS.



RESTful Web Service

RESTful Web Services (or RESTful web API) is an HTTP based net service. It will be used as an interface to receive, analyze and store the data. Processed data will be also shared, so they can be used to visualize battery parameters in the web app. The REST API will provides URL addresses as identifiers for particular operations. The device is meant to send acquired data in JSON format to a specific address, while the web app's task is to display them in a processed form.



Surroundings temperature readings

The existing prototype will be upgraded with the ability to read the temperature's surroundings. As before, the acquired data will be sent to the computing cloud for further analysis. Practically any temperature sensors on the market can be used to achieve this.

We recommend the use of DS18B20 sensor (see documentation), which works based on a 1-wire communication protocol.



Approximate device location readings

One other functionality is reading the approximate device (vehicle) location. It can be done in several ways, e.g. by installing an additional GPS module. However, the GSM module can be used just as well to determine approximate location in a similar way to mobile phones and A-GPS technologies. Using the Hayes language (AT commands) it is easy to download information about the current GSM network and the transmitters of GSM operators, whose signal is received by the modem. The LAC and CID parameters can be used to determine the approximate location.



Approximate device location readings

LAC (Location Area Code) – a unique location number that describes the set of base transmitters, merged into a group to optimize the signal quality.

CID (Cell ID) – also a unique number used to identify BTS stations in the LAC range.



Receiving and storing position, temperature and battery voltage data

The Dev's Kitchen participants' task will be to implement the method of receiving data from the device (with the set model) with a HTTP query.

Moreover, the received data should be saved into a database (DataFrames sheet) using the repositories described in the documentation. It will be necessary to establish a common model between the device and the REST API.



Analyzing data from the device with special attention to unusual occurrences

Thanks to the ADC converter built into the microcontroller and the resistor divider, it is possible to monitor the battery in real time. Based on this analysis we can conclude the current state of the battery, and comparing current data points with historical ones should bring us insight into its aging process. Situations which indicate accelerated aging or wear can be identified as well.



Text message notifications

The system's user should be able to monitor unusual occurrences via text messages. Adding a valid phone number for notifications should be done via the web app. To do this, the REST API service should share the query, which enables pairing the user's number with a specific device. Saving the number should result in sending the welcoming text message and sending another one every time something unnatural occurs.



Web app

The simple web app is an integral part of the „STARK” system and it's meant to comprehensively present the gathered data. REST API should share the queries enabling to perform all web app functionalities.

Thank you for your attention!

