The Journal of Open Source Software

Tenta: Remote and Real-Time Sensor Network Management

Felix Böhm [1¶], **Moritz Makowski** [1], **Patrick Aigner** [1], and **Jia Chen** [1¶]

1 Professorship of Environmental Sensing and Modeling, Technical University of Munich (TUM), Munich, Germany ¶ Corresponding author

DOI: 10.21105/joss.07311

Software

- Review
- Repository
- Archive

Editor: Kanishka B. Narayan

Reviewers:

- @abhishektiwari
- @sash19

Submitted: 20 June 2024
Published: 05 January 2025

Summary

In many domains, sensor networks have become indispensable to drive effective decisions: They monitor greenhouse gases and air pollutants (Dietrich et al., 2021; Wenzel et al., 2021), record volcanic activity (Werner-Allen et al., 2006), and track animal migrations (Zhang et al., 2004). However, when these networks are more than a few sensors in scale and/or their locations are difficult to access, manual inspection of sensors to retrieve their data and check their status gets prohibitively time-intensive. Simultaneously, scale renders the occurrence of sensor failures practically unavoidable. Failures that stay undetected lead to data loss (Bart et al., 2014; Tolle & Culler, 2005) and skewed results, and thus, potentially, to erroneous conclusions.

Tenta is a lightweight server application to manage sensor networks remotely and in real-time. The system receives and stores measurements transmitted by sensors and allows users to monitor sensors via an intuitive dashboard. Additionally, Tenta uses versionized configurations to adapt sensors to changes in requirements. Configurations are schemaless JSON documents that represent a sensor's state. They allow requests to sensors e.g. to update their measurement interval or even their software, as for instance implemented in Ivy (Makowski et al., 2024). Tenta associates each measurement with the sensor's currently active configuration, which results in highly reproducible datasets. Sensors communicate with Tenta over the widely adopted MQTT protocol. The system exposes a REST API for straightforward task automation. Data is stored in a time series-optimized PostgreSQL+TimescaleDB (PostgreSQL Global Development Group, 2024; Timescale, 2024) database.

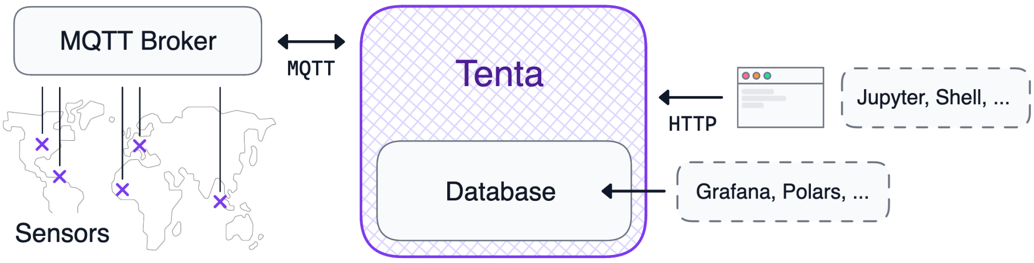


Figure 1: Tenta communicates with sensors via the MQTT protocol and exposes an intuitive dashboard and a REST API to manage them remotely. The time series-optimized database can be accessed directly to allow for elaborate processing and visualization e.g. with Polars or Grafana.

Starting in 2022, our research group deployed a network of 20 mid-precision CO2 sensors on rooftops in and around Munich (Aigner et al., 2024). To date, Tenta has reliably processed half a billion of this network's measurements as well as hundreds of configuration updates. Tenta's test suite currently stands at a statement coverage of 94%.

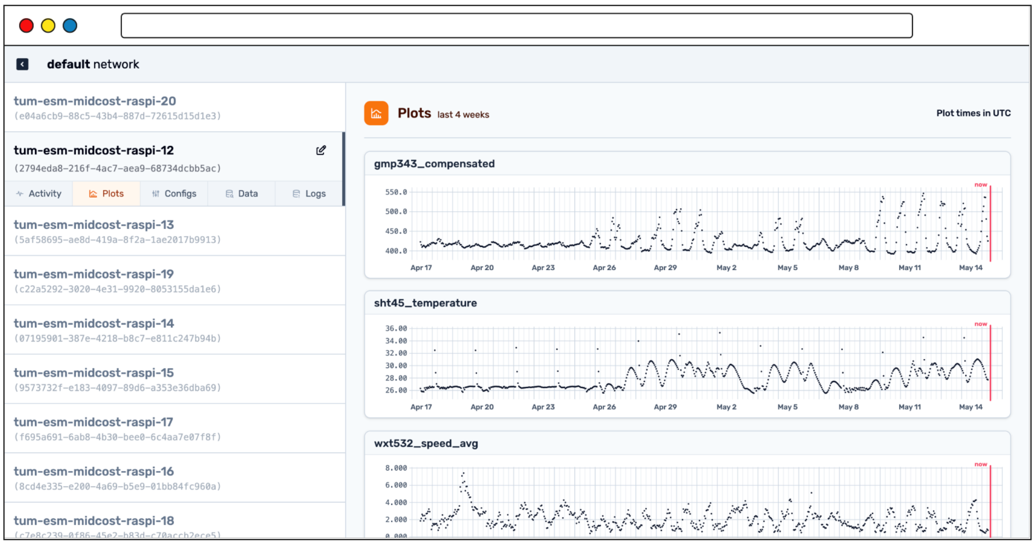
Statement of need

The IoT space is dominated by proprietary cloud services, e.g., from Amazon Web Services and Microsoft Azure. These platforms offer a holistic suite of message brokers, monitoring, device management, data storage, and analytics. Several research projects (Burri et al., 2019; Müller et al., 2020; Zweifel et al., 2021) use the sensors and infrastructure of the Swiss company Decentlab. Measurements are transmitted via LoRaWAN to gateways that relay them to Decentlab's infrastructure over the Internet. Other projects (Werner-Allen et al., 2006; Zhang et al., 2004) implement custom solutions e.g. based on flooding protocols. Smaller networks sometimes forgo connectivity entirely and instead rely on physical visits to the sites.

ThingsBoard (ThingsBoard, 2024) is an IoT platform to collect and visualize data and configure sensors via remote procedure calls. Multiple communication protocols are supported, including HTTP, MQTT, and CoAP. Furthermore, ThingsBoard allows data manipulation via so-called rule chains. A subset of ThingsBoard's functionality is available under an open-source license. Other open-source projects communicate only unidirectionally — from sensor to server — without the possibility of managing devices remotely: ThingSpeak (Lawlor, 2011) provides an HTTP API to store and retrieve data from IoT devices. Users can analyze and visualize data via the integrated MATLAB environment. The FROST Server (Schaaf, 2024) implements the OGC SensorThings API and provides both HTTP and MQTT interfaces to collect and query IoT data. ThingDirectory (Tavakolizadeh & Devasya, 2021) allows storing and querying IoT metadata (e.g. location, measurement type) via its REST API.

System design

Tenta follows the Unix philosophy and aims to be fast, flexible, and scalable. The project focuses on real-time sensor management and reliable data collection and storage. Tenta provides seamless interfaces, such that related tasks like data processing and custom visualizations can be accomplished with the array of great existing tools e.g. Polars (Vink, 2024) or Grafana (Ödegaard, 2024). Each of Tenta's features is designed to be optional. This makes it easy to connect the first sensor and allows Tenta to be used solely as a data sink for measurements on the one extreme or solely as a management system on the other. Figure 2 shows Tenta's dashboard with real-time charts of collected measurements.



default network

tum-esm-midcost-raspi-20
(e04a6cb9-88c5-43b4-887d-72615d15d1e3)

tum-esm-midcost-raspi-12
(2794eda8-216f-4ac7-aea9-68734dcbb5ac)

tum-esm-midcost-raspi-13
(5af58695-ae8d-419a-8f2a-1ae2017b9913)

tum-esm-midcost-raspi-19
(c22a5292-3020-4e31-9920-8053155da1e6)

tum-esm-midcost-raspi-14
(07195901-387e-4218-b8c7-e811c247b94b)

tum-esm-midcost-raspi-15
(9573732f-e183-4097-89d6-a353e36dba69)

tum-esm-midcost-raspi-17
(f695a691-6ab8-4b30-bee0-6c4aa7e07f8f)

tum-esm-midcost-raspi-16
(8cd4e335-e200-4a69-b5e9-01bb84fc960a)

tum-esm-midcost-raspi-18
(c7e8c239-0f86-45e2-b83d-c70accb2ece5)

Plots last 4 weeks

Plot times in UTC

Figure 2: Tenta's dashboard shows real-time charts of collected measurements, here with data from our research group's CO2 sensor network. Users can furthermore manage configurations and inspect measurements, logs, and status metrics via the dashboard.

Tenta imposes minimal restrictions on measurements and configurations. Their format can differ between sensors and can change over time without needing to notify the server beforehand, which allows Tenta to support sensor networks of almost arbitrary structure and simplifies software and hardware migrations on the sensor side. Individual sensors can even operate multiple different measurement formats at the same time. Despite this flexibility, Tenta persists measurements in a relational way, which keeps further processing straightforward.

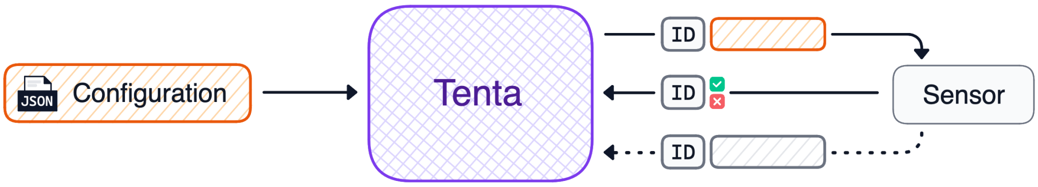


Figure 3: When a new configuration is created, it is assigned a unique identifier and relayed to the sensor over MQTT. The sensor positively or negatively acknowledges the configuration and can subsequently associate outgoing messages (e.g. measurements) with that identifier.

Configurations are schemaless JSON documents that represent sensor state. Figure 3 shows how Tenta assigns each configuration an incremental revision number before relaying it to the sensor. To increase observability, configurations are positively or negatively acknowledged. Subsequently, sensors can associate their outgoing messages (e.g. measurements) with the revision number of the relevant configuration.

Sensors communicate with Tenta over the widely adopted MQTT protocol. MQTT is designed for resource-constrained devices and consumes little bandwidth, which is vital for sensors connected via the mobile network. Client libraries are available in most programming languages, which maximizes flexibility with regard to sensor software and hardware.

Tenta leverages concurrency to achieve high throughput. Database and HTTP requests as well as MQTT message sending and receiving execute asynchronously. Tenta supports multiple sensor networks on a single instance as well as multiple users with different permissions.

Author contributions

FB: Designed and developed the server software, wrote the tests, wrote the documentation, wrote the manuscript; MM: Developed the dashboard, developed the client library running on the sensors, reviewed the manuscript; PA: Supervised the project, extracted the requirements, reviewed the manuscript; JC: Principle investigator, supervised the project, reviewed the manuscript;

Acknowledgment of financial support

This work was financially supported by the European Union's Horizon 2020 Programme as part of the ICOS Cities project (Pilot Applications in Urban Landscapes - Towards integrated city observatories for greenhouse gases - PAUL): Grant Agreement ID 101037319.

References

Aigner, P., Kübacher, D., Wenzel, A., Schmitt, A., Böhm, F., Makowski, M., Angleitner, M., Kürzinger, K., Laurent, O., Rubli, P., Grange, S., Emmenegger, L., & Chen, J. (2024). *Advancing urban greenhouse gas monitoring: Development and evaluation of a high-density CO2 sensor network in Munich*.

Bart, M., Williams, D. E., Ainslie, B., McKendry, I., Salmond, J., Grange, S. K., Alavi-Shoshtari, M., Steyn, D., & Henshaw, G. S. (2014). High density ozone monitoring using gas sensitive semi-conductor sensors in the Lower Fraser Valley, British Columbia. *Environmental Science & Technology*, *48*(7), 3970–3977. https://doi.org/10.1021/es404610t

Burri, S., Haeler, E., Eugster, W., Haeni, M., Etzold, S., Walthert, L., Braun, S., & Zweifel, R. (2019). How did swiss forest trees respond to the hot summer 2015? *Die Erde*, *150*(4), 214–229. https://doi.org/10.12854/erde-2019-420

Dietrich, F., Chen, J., Voggenreiter, B., Aigner, P., Nachtigall, N., & Reger, B. (2021). MUCCnet: Munich urban carbon column network. *Atmospheric Measurement Techniques*, *14*(2), 1111–1126. https://doi.org/10.5194/amt-14-1111-2021

Lawlor, L. (2011). ThingSpeak. In *GitHub repository*. GitHub. https://github.com/iobridge/thingspeak

Makowski, M., Böhm, F., Chen, J., & Aigner, P. (2024). *Ivy: A data acquisition system for distributed sensor networks supporting remote configuration and software updates* (Version 1.0.0). Zenodo. https://doi.org/10.5281/zenodo.14562882

Müller, M., Graf, P., Meyer, J., Pentina, A., Brunner, D., Perez-Cruz, F., Hüglin, C., & Emmenegger, L. (2020). Integration and calibration of non-dispersive infrared (NDIR) CO_2 low-cost sensors and their operation in a sensor network covering Switzerland. *Atmospheric Measurement Techniques*, *13*(7), 3815–3834. https://doi.org/10.5194/amt-13-3815-2020

Ödegaard, T. (2024). Grafana. In *GitHub repository* (Version 10.4.1). GitHub. https://github.com/grafana/grafana

PostgreSQL Global Development Group. (2024). PostgreSQL. In *GitHub repository* (Version 16.2.0). GitHub. https://github.com/postgres/postgres

Schaaf, H. van der. (2024). FROST server. In *GitHub repository* (Version 2.3.1). GitHub. https://github.com/FraunhoferIOSB/FROST-Server

Tavakolizadeh, F., & Devasya, S. (2021). Thing directory: Simple and lightweight registry of IoT device metadata. *Journal of Open Source Software*, *6*(60), 3075. https://doi.org/10.21105/joss.03075

ThingsBoard. (2024). ThingsBoard. In *GitHub repository* (Version 3.6.3). GitHub. https://github.com/thingsboard/thingsboard

Timescale. (2024). TimescaleDB. In *GitHub repository* (Version 2.14.2). GitHub. https://github.com/timescale/timescaledb

Tolle, G., & Culler, D. (2005). Design of an application-cooperative management system for wireless sensor networks. *Proceeedings of the Second European Workshop on Wireless Sensor Networks, 2005.*, 121–132. https://doi.org/10.1109/EWSN.2005.1462004

Vink, R. (2024). Polars. In *GitHub repository* (Version 0.20.17). GitHub. https://github.com/pola-rs/polars

Wenzel, A., Chen, J., Dietrich, F., Thekkekara, S. T., Zollitsch, D., Voggenreiter, B., Setili, L., Wenig, M., & Keutsch, F. N. (2021). Stand-alone low-cost sensor network in the inner city of munich for modeling urban air pollutants. *EGU General Assembly Conference Abstracts*, EGU21–15182. https://doi.org/10.5194/egusphere-egu21-15182

Werner-Allen, G., Lorincz, K., Ruiz, M., Marcillo, O., Johnson, J., Lees, J., & Welsh, M. (2006). Deploying a wireless sensor network on an active volcano. *IEEE Internet Computing*, *10*(2), 18–25. https://doi.org/10.1109/MIC.2006.26

Zhang, P., Sadler, C. M., Lyon, S. A., & Martonosi, M. (2004). Hardware design experiences in ZebraNet. *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, 227–238. https://doi.org/10.1145/1031495.1031522

Zweifel, R., Sterck, F., Braun, S., Buchmann, N., Eugster, W., Gessler, A., Häni, M., Peters, R. L., Walthert, L., Wilhelm, M., & others. (2021). Why trees grow at night. *New Phytologist*, *231*(6), 2174–2185. https://doi.org/10.1111/nph.17552