LATVIAN JOURNAL OF PHYSICS AND TECHNICAL SCIENCES
2017, N 3

DOI: 10.1515/lpts-2017-0016

TRANSITION FROM LEGACY TO CONNECTIVITY SOLUTION FOR INFRASTRUCTURE CONTROL OF SMART MUNICIPAL SYSTEMS

A. Zabasta, N. Kunicina, K. Kondratjevs

Riga Technical University,
12/1 Azenes Str., Riga, LV-1048, LATVIA
e-mail: Kunicina@latnet.lv

Collaboration between heterogeneous systems and architectures is not an easy problem in the automation domain. By now, utilities and suppliers encounter real problems due to underestimated costs of technical solutions, frustration in selecting technical solutions relevant for local needs, and incompatibilities between a plenty of protocols and appropriate solutions. The paper presents research on creation of architecture of smart municipal systems in a local cloud of services that apply SOA and IoT approaches. The authors of the paper have developed a broker that applies orchestration services and resides on a gateway, which provides adapter and protocol translation functions, as well as applies a tool for wiring together hardware devices, APIs and online services.

Keywords: *municipal systems automation, smart meters, SOA and IoT, wireless sensor networks.*

1. INTRODUCTION

Nowadays, most automation applications are supported on closed systems with limited capabilities to evolve, which can only be done at high costs. Collaboration between heterogeneous systems and architectures is not an easy problem to solve – especially in the automation domain. Legacy protocols, commercial off-the-shelf products and monolithic architectures often prevent setup, interoperability and dynamic reconfigurability of systems.

By now, utilities (heat suppliers, water suppliers, building service providers, etc.) and suppliers in their practice encounter real problems preventing development of smart meters due to underestimated costs of technical solutions, frustration in selecting technical solutions relevant for local needs, lack of technical competences and incompatibility among protocols used in existing and new equipment offered by suppliers.

Usually utility control systems interact and communicate with typical SCADA networks and Bus protocols (such as CAN, Profibus, IEEE-1394, standard Ethernet, etc). Modbus is often used to connect a supervisory computer with a remote terminal unit (RTU) in supervisory control and data acquisition (SCADA) systems. However, legacy systems typically provide little means for direct connectivity to access remote services over the internet.

Several frameworks have recently been created to solve problems of incompatibility. One of them is the concept of Industry 4.0 [1]. In Industry 4.0, the Internet of Things (IoT), Cyber-Physical Systems (CPS) and other concepts are to be used in order to break down the classical strict hierarchical approach of ISA-95 [2], replacing it with a more flexible approach without barriers and closed systems. The trend of applying the Industrial IoT (IIoT) technologies and specifically a Service-Oriented Architecture (SOA) to these systems requires changes in the philosophy applied to its development [3].

The Arrowhead project (ARTEMIS programme) [4] defined a framework for creating distributed industrial systems by collaborative networked embedded devices [5], [6]. This framework intends to integrate popular application layer protocols using a unique SOA. It aims at providing interoperability by facilitating the service interactions within closed or at least separated automation environments called local clouds.

The use of SOA enabled protocols such as CoAP [7], MQTT [8] and OPC-UA [9] allows for machine-to-machine (M2M) communication as well as interactions among end-users, sensor and actuator platforms. Additionally, CoAP addresses issues such as low-power access to resource-constrained devices and powerful scripting frameworks for service composition targeted CoAP.

The aim of the research is to provide evidence for practical implementation of the new generation of public utility network automated monitoring system, which complies with an industrial automation framework approach.

A smart service broker suitable for water distribution networks, district heating substations as well as temperature and humidity measurement in the offices and multi-apartment buildings has been developed and tested in real conditions. The broker resides on a gateway, which provides adapter and protocol translation functions. It applies a tool for wiring together hardware devices, APIs and online services and implements orchestration services for the cloud of public utility system. Thanks to enhancing of sensors and gateway devices, it becomes possible to design and implement a common architecture for public utility systems of a municipality.

The present research is conducted within the framework of the State Research Program NexIT "Sensor Networking and Signal Processing Applications in the National Economy".

2. ARCHITECTURE OF THE SMART MUNICIPAL SYSTEM-OF-SYSTEMS

Smart services rely on direct interaction between devices and web-based services, e.g., for real-time tracking of device condition from remote entities, such as maintenance and repair centres. Optionally, metering data are routed through customer SCADA or other automation systems, using existing technologies down to fieldbus systems. Therefore, the ways to interact and communicate with the device across multiple layers and hierarchies of networks, from a cloud service through the internet, GSM or other networks directly to the device, need to be supported.

The Arrowhead Framework uses a service-oriented approach to tackle interoperability and integrability issues within closed or at least separated automation environments called local clouds. It is done by facilitating and governing the service interactions between Application Systems using the Arrowhead Core Systems [10].

According to the Framework, a service can be a piece of information that is to be exchanged, e.g., sensor readout or actuating command. Arrowhead-compatible systems are supported by the services of the Core Systems when implementing their operational targets. The Core Systems themselves only help to establish and manage the connections, but the data exchange between application service producers and consumers is handled through the appropriate (even legacy) application protocols. There are three mandatory Core Systems that have to be present in a minimalistic local cloud; however, there are further, optional ones to support automation.

Figure 1 shows the "Smart Municipal SoS (System-of-Systems)" as a model comprising four application systems: water supply, district heating, building maintenance systems and electricity supply system. Three core systems operating as sharing systems for the local cloud of municipality systems provide authorization, service registry and orchestration services. Conditionally, this SoS can operate as a local cloud of several utility monitoring and control systems.

More detailed description of core services implemented within the Arrowhead project can be found in [4], [10]. Legacy systems such as client billing and inventory system comprise part of the overall municipal system, and to be integrated it is necessary to use protocol adapters; however, this is out of the scope of the present research.

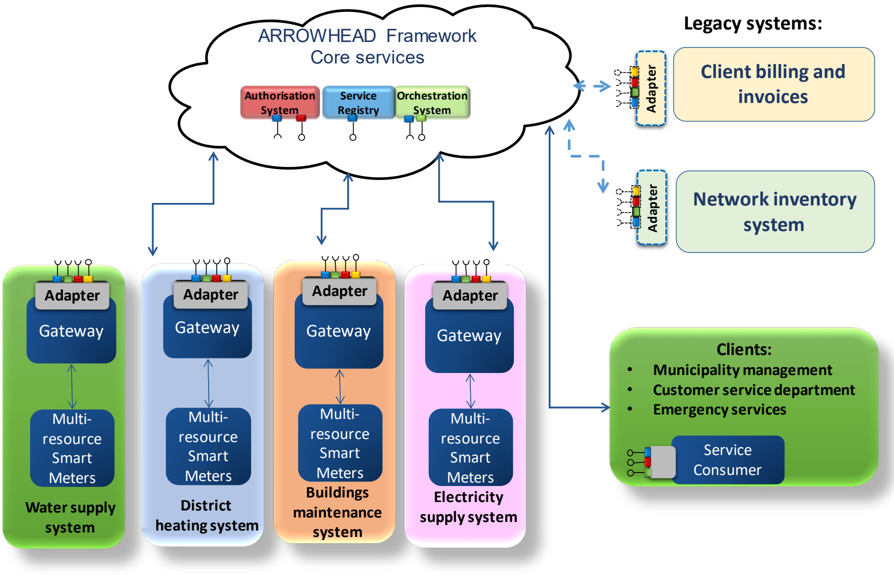

Fig. 1. A model view of smart municipal SoS.

3. SMART SERVICE BROKER

As an alternative to the web service implementation, SOA architecture based on a message queue with publishing and subscribing functionality was developed for Municipal SoS. Its implementation is based on the Message Queue Telemetry Transport (MQTT) protocol standardised in OASIS [11]. This machine-to-machine (M2M)/ Internet of Things connectivity protocol provides an extremely lightweight publishing/subscribing message transport with the benefits of [12], [13]:

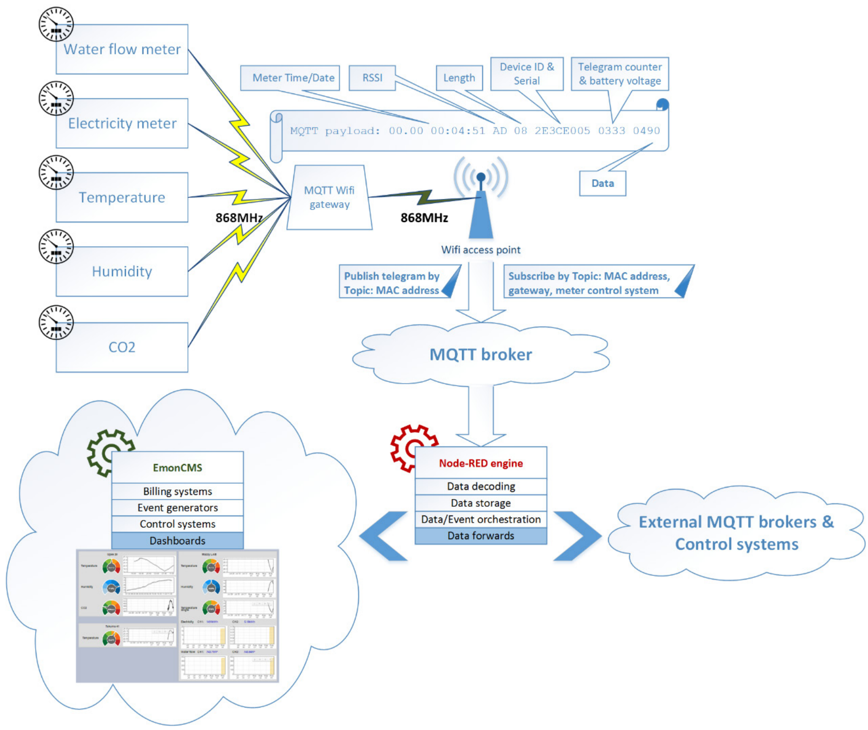

Fig. 2. MQTT based service broker.

- Extending connectivity beyond enterprise boundaries to smart devices;
- Offering connectivity options optimised for sensors and remote devices;
- Delivering relevant data to any intelligent, decision-making asset that can use it;
- Enabling massive scalability of deployment and management of solutions.

The MQTT protocol is based on the principle of publishing messages and subscribing to topics. MQTT messages are published by the provider device to topics at the message broker (see Fig. 2).

Consumer in turn signs up to receive particular messages by subscribing to topics at the broker. The messages broker takes over the role of the service registry and the authorisation system. Subscriptions can be explicit, limiting the messages that are received to the specific topic at hand, or they can use wildcard designators, such as a number sign (#), to receive messages for a variety of related topics [12].

For security reasons, encryption across the network will be handled with SSL/TLS independently of the MQTT protocol itself. The orchestration system will be the same as for the web service solution: a business process engine based on the Business Process Model and Notation (BPMN).

The main advantage of this solution compared to the web service implementation is its “push concept”, which does not require open input ports at the device owner side. It also enables a comparatively simple-to-implement solution based on proven methods and existing tools.

4. MULTISOURCE SENSORS AND GATEWAYS

The implemented system consists of a set of custom designed smart metering units providing different types of telemetry data and control applications: sound measurements – sampled averaged values of noise in an environment; pressure measurements – water supply systems; water flow – water flow for domestic meters; electricity meters – impulse interface for common electricity meters; temperature meters; humidity meters; CO_2 meters; strain meters – usable for deformation registration and scale applications. Such a list of smart meters is able to represent the majority of utility control systems operating at the territory of city or municipality.

The proposed solution for communication as part of public utility system contains a multi-interface modular platform with two main node components. The first one – a metering node – connects to a meter via switchable/selectable interfaces (current loop, IEC1107 optical interface etc.). Inter-system communication is possible using selectable interface modules (e.g., IEEE 802.3, ISM radio interface, GSM/GPRS). The second one – a gateway node – has a selectable inter-system and backend communication interface architecture. The gateway node provides requests, readout pre-processing, secure data delivery, queuing etc. The gateway nodes consist of a radio module and 802.11b/g/n Wi-Fi module that integrates a microcontroller (ESP8266). The Wi-Fi module supports both station (Wi-Fi client mode) and access point modes.

The web interface shows information about the gateway and allows performing manual Wi-Fi client configuration options. An automated SSID scan can also be initiated to scan for available Wi-Fi networks; the web interface then allows inputting authorisation data and parameters, such as MQTT broker hostname/IP address and authorisation data to proceed with metering device telegram transport using Wi-Fi connection password. The gateway automatically redirects the connecting client device to a configuration web interface.

5. SERVICE ORCHESTRATION

Orchestration in the context of SOA can be viewed as the system that supports the establishment of all other systems through providing coordination, control and deployment services. The orchestration system is a central component of the Arrowhead Framework as in any SOA-based architecture [14]. Orchestration is used to control how systems are deployed and in what way they should be interconnected.

We deployed a Node-RED [15] as an orchestration tool for wiring together hardware devices, APIs and online services in a new convenient way. At the heart of Node-RED, there is a visual editor allowing complex data flows to be wired together with only little coding skills. The main functionality of Node-RED is to decode and route MQTT smart metering data to further service orchestration or use in external services as customer billing or monitoring systems.

Multiple operations are implemented to decode the payload and forward it to a data storage and visualisation service using the IoT approach:

- The first step is to define MQTT source, which is a gateway. Gateways post the received metering data to topics based on their MAC address that also serves as a configuration service by subscribing to MAC/configuration. Corresponding QoS levels can be specified for both published and subscribed topics.
- The next node is "SERIAL+VALUE" – this creates an array of elements from the initial payload (telegram), then each element of the array is processed separately depending on the needed output or post-processing. A new object is created by dividing name/value pairs, where the crucial elements are: serial number that identifies the data type and coding format, and the unique device ID, which makes the metering devices distinct. The second value is the data block needed for decoding procedures.
- Double output is formed with two new MQTT messages containing separated data from each sensor, with measurement type topic and device unique serial number.

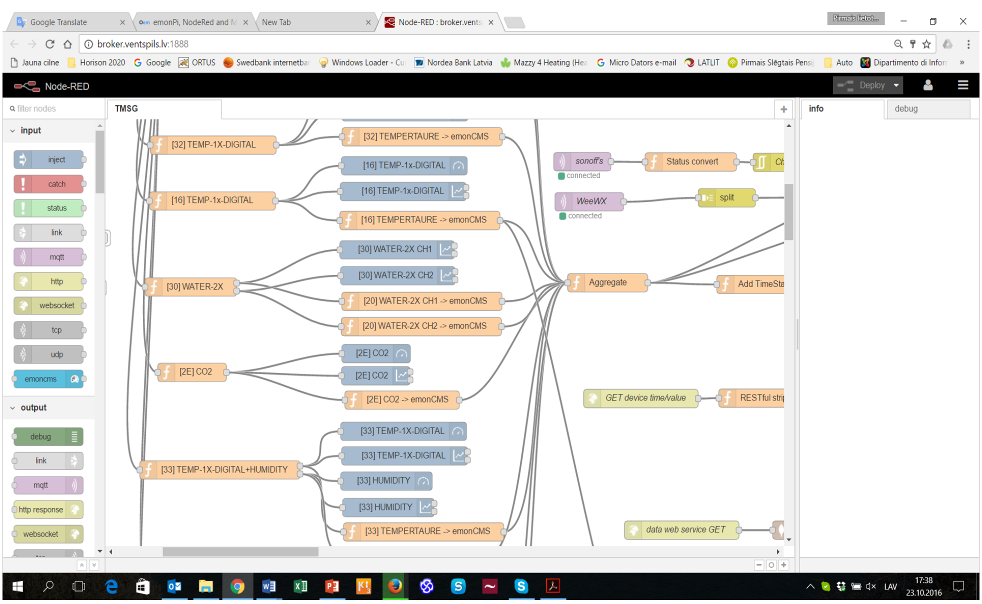

Fig. 3. Services orchestration implementation of MQTT based service broker using Node-RED flows.

- For further processing in customer billing systems and monitoring, EmonCms [16] code base is used. JSON serialized API calls are prepared using a separate node "(33) TEMPERTAURE -> EmonCMS" and "(33) HUMIDITY -> EmonCms". Data binding to EmonCms is performed using a shared API key. For easier identification of different sensor types the type number is included in the node description in brackets, for example, the node (33) is a single digital temperature sensor with a humidity sensor.
- From the example above a type+device_id object name is generated. After data delivery to EmonCms, these data objects are discovered as inputs. Inputs are generic variables that can be manipulated using EmonCms processing engine. Users can customise data transformation, delta offsets, calibration, scaling etc.
- Typically, the broker does not save MQTT data; this can be done by defining a flow to data storage, such as NoSQL MongoDB. In this application, data storage is defined in EmonCms with the "Log to Feed" processor.
- Before saving information for analysis and representation, several pre-calculations, such as impulse value adjustments for water and electricity meters and offset values, to estimate the actual meter readout value should be done.
- From the inputs that are pre-processed and stored, a new value "Feed" is created. Feeds are data streams on demand, which can be used for time series analysis, statistics, and data export functionality.

For the custom application, a visualisation dashboard has been created using EmonCms to show advanced widget use and smart metering data display.

The MQTT published data are self-describing, where it is up to the systems designer to choose the topic structure. In most cases, the first topic is the location or systems identifier. For easier filtration and data processing the subtopic is the unique clients' identificator – in this case – client 0001. The type of general metering is defined (water) followed by the type of measurement and the measurement sensor unit.

Special wildcards can be used to filter groups of data. In the given sample, the "+" wildcard would include all variations of the client unique ID; therefore, by such filtering data are available to all clients for a selected and authenticated site. The dash symbol at the end is a multilevel wildcard that matches all underlying topics – in this case all water flow meter units.

Example returned value from a request for the latest data of the virtual energy of water heating feed (ID=172: Request from 12.09.2016 to 13.09.2016 at an interval of 5 minutes from the feed ID=172: http://broker.ventspils.lv:9990/emoncms/feed/data.json?id=172&start=1473638400&end=1473724800&interval=5

Returned object JSON serialized:

1473638000,-21851.932456474],[1473643000,-
21851.932456474],[1473648000,-
21851.932456474],[1473653000,-
21851.932456474],[1473658000,-
21851.932456474],[1473663000,-21851.932456474],etc.

RESTful services can be implemented on demand from the EmonCMS API by encapsulating the API request into a restful call or by direct definition of a request in form of a RESTful HTTP request. Example requests: JSON export using API (19.09.2016 00:00:00 to 19.09.2016 01:00:00 using intervals of 5 minutes):

Request: http://broker.ventspils.lv:9990/emoncms/feed/data.json?id=130&start=1474232400000&end= 1474236000000&interval=5

Response: [[1474232648000,916.857],[1474233287000,916.857],[1474233925000,916.857],[1474234563000,916.857],[1474235202000,916.857],[1474235840000,916.857]]

Example of JSON request time of latest value: http://broker.arrowhead.bitdev.lv: 9990/ emoncms/feed /timevalue.json?id=132

Response: {"time":"1476434485","value":"904.382"}. Unix timestamp 1476434485 corresponds to GMT: Fri, 14 Oct 2016 08:41:25 GMT

6. CONCLUSIONS

Since public utilities in municipalities pursue automation of processes but install and maintain various systems, the issue of the maintenance cost optimisation becomes very crucial, and particularly, when the maintenance costs result in the growth of tariffs for public services. A practical implementation of the Arrowhead Framework approach for optimisation of municipal public utility systems demonstrates promising opportunities for system maintenance cost reduction. MQTT based broker enables smart service creation and collaboration among different utility monitoring and control systems.

ACKNOWLEDGEMENTS

The research has been supported by the Latvian Ministry of Education and Science within the framework of Latvian State Research Program "The Next Generation of Information and Communication Technologies (NexIT)" (2014–2017).

REFERENCES

1. Lee, J., Bagheri, B., & Kao, H.-A. (2015). A cyber-physical systems architecture for industry 4.0-based manufacturing systems. *Manufacturing Letters*, *3*, 18–23.
2. Scholten, B. (2007). *The Road to Integration: A Guide to Applying the ISA-95 Standard in Manufacturing*. ISA
3. Da Xu, L., He, W., & Li, S. (2014). Internet of things in industries: A survey. *IEEE Transactions on Industrial Informatics, 10*(4), 2233–2243.
4. Arrowhead Framework Wiki. Retrieved 12 September 2016 from https://forge.soa4d.org/ plugins/media wiki/wiki/arrowhead-f/index.php/Main Page
5. Blomstedt, F., Ferreira, L.L., Klisics, M., Chrysoulas, C., de Soria, I.M., Morin, B., Zabasta, A., Eliasson, J., Johansson, M., & Varga, P. (2014). The arrowhead approach for SOA application development and documentation. In IECON 2014 – 40th Annual Conference of the IEEE Industrial Electronics Society, 29 October – 1 November 2014 (pp.

2631–2637). USA: Dallas.

6. Varga, P., Blomstedt, F., Lino Ferreira, L., Eliasson, J., Johansson, M., Delsing, J., & Martinez de Soria, I (2016). Making system of systems interoperable – The core components of the arrowhead technology framework. *Journal of Network and Computer Applications*.
7. Shelby, Z., Hartke, K., & Bormann, C. (2014). *The constrained application protocol (CoAP)*.
8. Holm, T., Christiansen, L., Gring, M., Jger, T., & Fay, A. (2012). ISO 15926 vs. IEC 62424 - Comparison of plant structure modelling concepts. In *Proceedings of 2012 IEEE 17th International Conference on Emerging Technologies Factory Automation* (ETFA 2012), 17–21 September 2012 (pp. 1–8). Krakow: Poland.
9. Delsing, J., Rosenqvist, F., Carlsson, O., Colombo, A. W., & Bangemann, T. (2012). Migration of industrial process control systems into service oriented architecture. In IECON 2012, 25–28 October 2012 (pp. 5790–5796). Canada: Montreal.
10. Delsing, J. (Ed.) (2016). *IoT Automation: Arrowhead Framework*. CRC Press, Taylor & Francis Group.
11. OASIS Message Queuing Telemetry Transport (MQTT) Technical Committee Charter. Retrieved 16 October 2016 from https://www.oasisopen. org/committees/mqtt/charter.php.
12. Lampkin, V., Leong, W. T., Olivera, L., Rawat, S., Subrahmanyam, N., & Xiang, R. (2012). *Building Smarter Planet Solutions with MQTT and IBM WebSphere MQ Telemetry.* IBM.
13. MQ Telemetry Transport. Retrieved 16 October 2016, from http://mqtt.org/
14. Varga, P., & Hegedus, C. (2015). Service interaction through gateways for inter-cloud collaboration within the arrowhead framework conference. In *5th International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronics Systems (Wireless VITAE),* 13–16 December 2015. India: Hyderabad. Available at https://www.researchgate.net/publication/287878312.
15. Open Energy Monitor Project. Node-RED. Retrieved 16 October 2016, from http://guide.openenergymonitor.org/integrations/nodered/.
16. Open Energy Monitor Project. Web-app for processing, logging and visualizing energy, temperature and other environmental data. Retrieved 16 October 2016, from https://emoncms.org/.

PĀREJA NO MANTOJUMU RISINĀJUMIEM UZ INTELEKTUĀLO MIJIEDARBĪBU STARP KOMUNĀLAS SAIMNIECĪBAS INFRASTRUKTŪRAS KONTROLES SISTĒMĀM

A. Zabašta, N. Kuņicina, K. Kondratjevs

Kopsavilkums

Nodrošināt sadarbību starp neviendabīgām sistēmām un arhitektūrām nav viegls uzdevums it sevišķi automatizācijas jomā. Līdz šim komunālās saimniecības uzņēmumi un aparatūras piegādātāji sastopas ar reālām problēmām sakarā ar tehnisko risinājumu kļūdaino izmaksu aprēķiniem, un grūtībām, izvēloties tehniskos risinājumus, kas atbilst vietējām vajadzībām, kā arī ar nesavietojamību starp daudzveidīgiem protokoliem un dažādu sistēmu ražotāju aizsargātiem interfeisiem.

Šis darbs prezentē pētījumu par intelektuālo komunālas saimniecības sistēmu arhitektūras izstrādi vietējā servisu mākonī, kas izmanto SOA un IoT pieeju. Brokeris, kas pielieto orķestrācijas pakalpojumus, balstās uz vārtejas, kas nodrošina starpsistēmu adaptera un protokolu tulkošanas funkcijas, kā arī uz Node-RED programmatūras, kas vieno kopā dažādas mērīšanas ierīces, APIs un tiešsaistes servisus. Node-RED rīks dekodē, apstrādā, un pārsūta MQTT viedo mērījumu datus turpmākai servisu orķestrēšanai, un to izmantošanai ārējos pakalpojumos, piemēram, klientu rēķinu izveidošanai, vai kā kontroles sistēmu ievadus. Izmantojot EmonCms rīku, ir izstrādāts viedās uzskaites un datu vizualizācijas displejs.

Arrowhead ietvara pieejas praktiska īstenošana, lai optimizētu pašvaldības komunālo sistēmu uzraudzību un kontroli, demonstrē daudzsološas iespējas sistēmu uzturēšanas izmaksu ietaupīšanā.

12.04.2017.