Towards Transparency of IoT Message Brokers

Milan Markovic[†], David Corsar[†], Waqar Asif[*], Peter Edwards[†], and Muttukrishnan Rajarajan[*]

[†]Computing Science, University of Aberdeen,
Aberdeen, AB24 5UA
[*]School of Engineering and Mathematical Sciences, City, University of London,
London, EC1V 0HB
{milan.markovic,dcorsar,p.edwards}@abdn.ac.uk
{waqar.asif,r.muttukrishnan}@city.ac.uk [⋆]

Abstract. In this paper we propose an ontological model for documenting provenance of MQTT message brokers to enhance the transparency of interactions between IoT agents.

Keywords: IoT · MQTT · Provenance

1 Introduction

The Internet of Things (IoT) enables multiple heterogeneous devices and applications to interact with each other using the Internet as a common communication infrastructure. However, these devices bring with them a new set of problems such as security and user data/identity privacy [1]. The concurrent operation of such devices leads to a high risk of data breach where a device capable of complex computations can launch an active or a passive attack in a network running weak security protocols [2]. Alongside this, the increasing interest of IoT users in data privacy and new regulations for protecting personal data such as the General Data Protection Regulation[1] and the Safe Harbor Framework[2] necessitate greater transparency of IoT systems. This includes user-accessible information on processes utilising the data generated/obtained through IoT devices.

We argue that transparency of interactions between IoT devices (e.g. exchanging of messages) is a critical enabler to support IoT device accountability, privacy, and data quality assessments. Here, the W3C recommendation PROV [5] could provide means to document causal relationships between agents (i.e. things, data consumers, etc.), activities they perform (e.g. sensing, relaying messages), and data entities used and generated. In addition, by documenting the

[⋆] The work described here was funded by the award made by the RCUK Digital Economy programme to the University of Aberdeen (EP/N028074/1) and City, University of London (EP/N028155/1).

[1] https://eur-lex.europa.eu/legal-content/en/LSU/?uri=CELEX3A32016R0679

[2] https://www.ftc.gov/tips-advice/business-center/privacy-and-security/u.s.-eu-safe-harbor-framework

intended actions of agents (i.e. plans) we could also support audit of IoT components in terms of system capabilities and deviations in behaviour.

In this context, message brokers implementing the MQTT standard[3] are commonly used to network groups of IoT devices and software agents[4]. In such networks all communication between clients can be inspected by auditing the connected message brokers. At the same time, malfunctions, misconfigurations or the limited capabilities of message brokers (e.g. not detecting abnormal behaviour such as repeated failed authentication attempts) pose significant security and privacy risks that may result in data loss or breach of data sharing permissions. Provenance records documenting the intended and actual behaviour of message brokers could support discovery of such issues. For example, a provenance query could reveal a list of all agents that had access to a redistributed message which can be checked against a user's policy for data sharing. Further queries could also identify messages that are not being forwarded by the broker (i.e. plans not executed in full), frequent attempts at unauthorised client subscriptions, or clients that are frequently disconnected due to their inactivity without properly closing their connections.

In the remainder of the paper we introduce the MQTT-PLAN ontology, designed to define plans describing the intended actions of brokers upon receipt of different types of MQTT control packets. This can then be used to annotate retrospective provenance records of the broker's actual behaviour, identifying correspondances between the retrospective entities and activities and concepts in the plan. We conclude with a discussion of outstanding challenges and outline our future work.

2 The MQTT-PLAN Ontology

MQTT-PLAN[5] defines a vocabulary extending PROV-O[6] and P-PLAN [3] to describe high level abstract plans associated with MQTT brokers; and their corresponding execution traces. The ontology captures information that could be found by inspecting individual MQTT control packets (e.g. message topics) and other information maintained by a broker (e.g. the identity of clients subscribed to receive messages published to each topic, and reasons for client disconnections). In P-PLAN, plans are modelled as sets of variables serving as inputs and outputs of steps. Execution traces are then described using the concepts *p-plan:Entity* and *p-plan:Activity*[7] which are linked to the corresponding plan via *p-plan:correspondsToVariable* and *p-plan:correspondsToStep*. This approach

[3] http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html

[4] MQTT is a publish/subscribe messaging transport protocol for a client-server communication. The protocol specifies a set of control packets that govern the communication between the client and the message broker residing on a server.

[5] http://w3id.org/mqtt-plan

[6] https://www.w3.org/TR/prov-o/

[7] Subclasses of *prov:Entity* and *prov:Activity*.

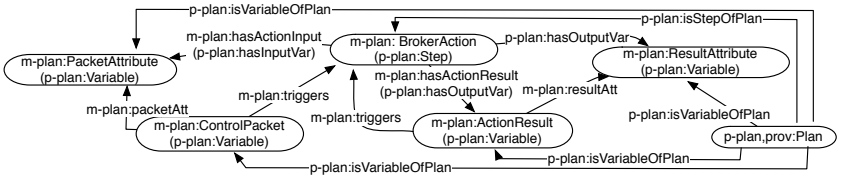

Fig. 1. Main MQTT-PLAN concepts modelled as subclasses of P-PLAN concepts.

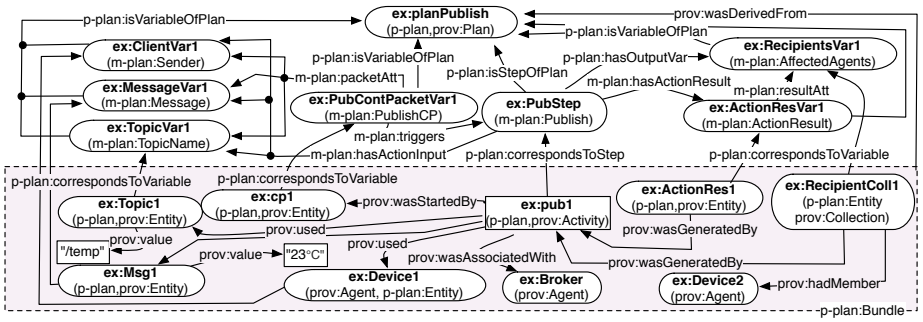

Fig. 2. An example provenance record describing a broker's plan for re-publishing received messages to other clients, and a record of the corresponding execution trace.

allows for a separation of the abstract plan description from individual execution traces describing instances of enacted processes and associated data. As a result, the interpretation of retrospective provenance is bound to the corresponding plan description. Fig 1 illustrates the main MQTT-PLAN concepts[8]. A step *m-plan:BrokerAction* can be triggered by *m-plan:ControlPacket* and its more descriptive subtypes such as *m-plan:PublishCP*, *m-plan:SubscribeCP*, etc. Subtypes of *m-plan:BrokerAction* such as *m-plan:Subscribe*[9], *m-plan:Publish*[10], *m-plan:Disconnect*[11] are also defined. Control packets can be associated with *m-plan:PacketAttribute*(s) such as *m-plan:TopicName* and *m-plan:Message*. Using property chain axioms associated with *m-plan:hasActionInput* such attributes are inferred as input variables of the *m-plan:BrokerAction*. A broker action can produce an *m-plan: ActionResult* variable, which describes a results object and can also trigger another *m-plan:BrokerAction* step. Such results can be associated with attributes *m-plan:Target*, *m-plan:Reason* and *m-plan:CompletionStatus*. Subtypes of *m-plan:Target*, namely *m-plan:AffectedAgent* and *m-plan: AffectedA-*

[8] MQTT-PLAN concepts are described with the prefix m-plan.

[9] The client sending a control packet triggering this action should be registered to receive messages published under the requested topics.

[10] A message specified in the control packet triggering this action should be forwarded to clients subscribed to the topic under which it was published.

[11] This action should close the connection between a client and a broker.

gents define variables which can be instantiated via a retrospective provenance record to describe either a single or group of agents affected by the activity instance corresponding to *m-plan:BrokerAction*. Fig 2 illustrates an example plan describing re-publishing of messages by a message broker and a corresponding execution trace. In this example, a message containing a temperature reading was published under the topic "/temp" by the client *ex:Device1* and was forwarded to the client *ex:Device2* by the agent *ex:Broker*.

Similarly, *m-plan:CompletionStatus* and *mplan:Reason* can be instantiated via the retrospective provenance record to determine whether the broker could complete the activity and the reason if this was not possible. For example, a publish control packet could trigger a publish action which could not complete due to the client being denied access to the topic. A result of this action could also trigger disconnection of the client.

3 Discussion and Future Work

In order to keep the vocabulary lightweight, the initial version of the ontology does not cover all of the functionalities specified in the MQTT standard. These include: quality of service tracking, handling of will messages, retaining of messages by the broker, session flags. Username and password flags are also not captured explicitly for security reasons. However, the ontology enables modelling of various plans describing single or multiple broker actions interlinked with their input and output variables. As part of our future work we aim to create a repository of common representations of broker's plans described using MQTT-PLAN to generate further community discussions about their use. We will also evaluate how the lack of support for conditional branches impacts on modelling such plans. Finally, traffic managed by message brokers presents scalability challenges. However, our previous work [4] demonstrated a possible approach using linked data streams. We are currently exploring a means for capturing provenance by extending an open source MQTT message broker in order to evaluate the potential of a stream-based approach for consuming such data.

References

1. Asif, W., Rajarajan, M., Lestas, M.: Increasing user controllability on device specific privacy in the internet of things. Computer Communications **116**, 200–211 (2018)
2. Bertino, E., Islam, N.: Botnets and internet of things security. Computer **50**(2), 76–79 (2017)
3. Garijo, D., Gil, Y.: Augmenting prov with plans in p-plan: Scientific processes as linked data. In: Proc. of the 2nd Int. Workshop on Linked Science 2012 (2012)
4. Markovic, M., Edwards, P.: Semantic stream processing for iot devices in the food safety domain. In: Posters&DemosSEMANTiCS 2016 and SuCCESS'16 Workshop (Leipzig, Germany, 2016)
5. Moreau, L., Groth, P., Cheney, J., Lebo, T., Miles, S.: The rationale of prov. Web Semantics: Science, Services and Agents on the World Wide Web **35**, 235–257 (2015)